REALTIME DATA MINING APLICADO A LA PREDICCIÓN DE ÍNDICES DE BOLSA INCLUYENDO SOCIAL MEDIA ANALYTICS

ANDRÉS FERNANDO FUENTES MEDINA

DIRECTOR: KARINA GIBERT OLIVERAS

DEPARTAMENTO DE ESTADÍSTICA E INVESTIGACIÓN OPERATIVA

MÁSTER EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

2017 JULIO 06

AGRADECIMIENTO

A mi directora, Karina Gibert Oliveras, por la orientación y el apoyo brindado durante el desarrollo del presente trabajo.

A Gaby, por su paciencia y apoyo durante estos meses de estudio.

RESUMEN

El presente trabajo tiene que ver con la obtención de modelos predictivos para la bolsa de valores que funcionen en intervalos cortos de tiempo. Además de trabajar con los precios que tienen las acciones en determinado instante, se incorporan los mensajes de la red social Twitter que tienen relación con dichas acciones y así verificar su impacto. Inicialmente se describe el estado del arte, los conceptos y herramientas utilizadas a lo largo del proyecto, ya que por su naturaleza interdisciplinar incorpora temas como el data mining, machine learning, sentiment analysis y computación distribuida.

Se explica la metodología usada, que va desde la extracción de datos hasta la presentación y evaluación de resultados. Se distingue dos partes fundamentales: análisis con datos históricos y análisis con datos en tiempo real. Se describe para ambas partes el preprocesamiento de datos, ya que estos por ser de distinta naturaleza se les aplica criterios diferentes, entre los que se trata la reducción de dimensión y el sentiment analysis. Para la obtención de los modelos predictivos se aplican tres métodos distintos de machine learning, se compara y se evalúa los resultados para finalmente escoger los mejores.

Con la información extraída en los pasos previos se lleva a cabo la implementación en tiempo real mediante Apache Spark, para hacer predicciones en tres intervalos distintos de tiempo. Tomando en cuenta las consideraciones de trabajar con un flujo constante de datos se describe una aplicación que optimiza el tiempo necesario para la evaluación de este tipo de modelos en ambiente de pruebas. Los resultados obtenidos se comparan con los de los modelos de datos históricos.

El resultado final del proyecto es la descripción de la arquitectura e implementación de un sistema para evaluación de modelos predictivos de acciones bursátiles con datos en stream. Las conclusiones se enmarcan en la verificación de agregar información de redes sociales a los clásicos modelos predictivos que toman en cuenta solamente precios de acciones, así como también el hecho de trabajar con información en tiempo real.

CONTENIDO

TABLAS

FIGURAS

1. INTRODUCCIÓN

“El reciente episodio de la aerolínea con uno de sus pasajeros en un vuelo sobrevendido en Chicago causó indignación en redes sociales y le provocó una caída de 4% en la Bolsa de Valores de Nueva York” [1]. Así empezaba la noticia haciendo referencia al incidente que se hizo conocer en la red social Twitter a través de la publicación de un vídeo en donde se mostraba la expulsión de un pasajero de la aeronave, en contra de su voluntad y de forma humillante, lo que desencadenó cientos de mensajes de rechazo a la mencionada aerolínea, y esto a la vez, una disminución en el precio de sus acciones.

En este sentido, el éxito de un inversionista está en saber con mayor certeza cuándo comprar o cuándo vender, basándose en su intuición, experiencia, el seguimiento minucioso de las variaciones de los precios de acciones y un análisis exhaustivo de las noticias que hablan sobre esas acciones, que generalmente son compartidas y comentadas a través de las redes sociales por miles de usuarios en todo el mundo. Lo importante es poder analizar con la mayor rapidez toda esta información y actuar antes que el resto, para lo cual se debe ir más allá de la capacidad humana.

Las herramientas y algoritmos informáticos usados en este campo se han centrado principalmente en analizar el contenido de la información que se publica constantemente en Internet y ofrecer una respuesta rápida para la toma de decisiones. No se trata solamente de hacer predicciones para los siguientes días, pues debido a la rapidez con la que se hacen actualmente las transacciones en la bolsa se busca resultados en el orden de los minutos o segundos [2].

El análisis de las fluctuaciones de precios de acciones en la bolsa es una tarea importante de minería de texto [3] y puede ser visto también desde la perspectiva de Big Data, por el volumen de información que puede llegar a procesarse, y también por la velocidad a la que ésta fluye.

1.1. Motivación

La principal motivación al desarrollar este trabajo ha sido el interés en aplicar la minería de datos a flujos de información permanente mediante el uso de Apache Spark y sus librerías de machine learning y streaming. Como lo es también trabajar en un proyecto interdisciplinar para aplicar los conocimientos en machine learning, sentiment analysis y computación distribuida.

Por otra parte, es muy común que cada día se encuentre en Internet o en los diarios información sobre índices financieros o empresas que cotizan en la bolsa de valores, por lo que, entender su funcionamiento y de qué forma es afectada por las redes sociales ha hecho que el proyecto sea un reto interesante.

1.2. Objetivos

En este trabajo se intentará modelar el comportamiento de los índices bursátiles en función de las fluctuaciones de las acciones y los mensajes de Twitter que se generan asociados a estas fluctuaciones. De forma general se analizarán dos puntos:

- El impacto de incluir información de redes sociales en el modelo clásico de relación entre acciones e índice bursátil.
- La ventaja de actualizar el modelo en tiempo real.

Para ello, se cubrirán los siguientes objetivos:

- Diseñar e implementar una herramienta que recupere los datos de acciones bursátiles en tiempo real de alguna fuente que los proporcione.
- Identificar el modelo de predicción a utilizar. Se usarán herramientas de modelización estadística conocidas, como la regresión o la clasificación. En este punto se trabajará con datos históricos y se compararán varias aproximaciones: Predicción usando solo datos históricos de la propia accion, predicción usando también los datos históricos de otras acciones del mercado y predicción incluyendo información procedente de Twitter.

Para introducir la información de Twitter en el modelo de predicción es necesario cubrir adicionalmente los siguientes objetivos:

- Determinar las características de los tweets que se quieren recuperar. Diseñar e implementar la herramienta que recupere los tweets requeridos.
- Realizar un sentiment analysis de los tweets, así como un proceso adicional de feature extraction que caracterice el contenido de los tweets en forma de un conjunto de indicadores compatibles con los formatos habituales en análisis de datos.
- Analizar de forma comparativa de todos los modelos predictivos construidos para seleccionar el mejor modelo.

- Evaluar las posibilidades de utilizar el mejor modelo identificado en un entorno de tiempo real de predicción en base a data streams.
- Diseñar e implementar la herramienta de predicción en tiempo real, conectando los componentes construidos en los objetivos previos.
- Validar y testear la herramienta diseñada.

1.3. Estructura de la memoria

La presente memoria comprende ocho capítulos en donde secuencialmente se irá trabajando en cada uno de los objetivos.

El capítulo dos habla del estado del arte en data mining y sentiment analysis, además de estudios y propuestas similares en la predicción de acciones e índices en la bolsa. También se incluirá brevemente una breve descripción de varias técnicas utilizadas en los temas que se ha mencionado.

El capítulo tres hace un resumen de los principales conceptos y herramientas que se usarán a lo largo del proyecto, entre ellas están: bolsa de valores, Twitter, Python y Apache Spark.

El capítulo cuatro muestra la metodología del proyecto. Se indicarán los pasos a seguir para alcanzar los objetivos planteados.

El capítulo cinco tiene que ver con la extracción de datos y su preprocesamiento, que incluye el sentiment analysis de los tweets.

El capítulo seis se involucra el trabajo con datos en batch y se crean varios modelos de predicción para finalmente escoger uno, el que mejores resultados genere.

El capítulo siete, por otra parte, involucra el trabajo con datos en stream y se aplica las herramientas disponibles en la generación de un modelo que se actualice continuamente. Al final se evalúan los resultados y se compara con el modelo con datos en batch.

El capítulo ocho incluye las conclusiones y el trabajo futuro que quedará planteado a partir de este proyecto.

2. ESTADO DEL ARTE

Obtener un modelo que prediga el precio de acciones es una área que ha sido estudiada durante varios años, y a pesar de que la Hipótesis del Mercado Eficiente sugiere que definitivamente las fluctuaciones en la bolsa de valores no se pueden predecir, se sigue en la búsqueda de los atributos y el algoritmo que ofrezca el mejor resultado. El análisis de la predicción se lo aborda desde dos perspectivas: técnica o fundamental. El análisis técnico sugiere que se utilice solamente precios históricos, en cambio el fundamental incorpora información de noticias, redes sociales y demás medios de comunicación que hagan mención al mercado bursátil [4].

Un ejemplo de análisis técnico en donde se intenta predecir la dirección de la variación en el precio de una acción en base a los cinco precios previos se analiza en [5], se utilizan varias técnicas de clasificación como Regresión Logística, LDA, QDA, KNN con resultados solamente un diez por ciento mejores a los obtenidos al lanzar una moneda.

El análisis fundamental es el más estudiado actualmente, técnicas de sentiment analysis aplicadas a microblogs como en [6] inician con la clasificación de información financiera o no financiera, para luego clasificarla nuevamente entre la que transmite sentimientos positivos o negativos, que posteriormente será agregada al dataset de acciones. Varios estudios se enfocan en analizar solamente los portales de noticias que hacen mención a las compañías que cotizan en bolsa [7] [8]. Otros por el contrario, utilizan enfoques híbridos, agregando información de redes sociales, en especial Twitter [4] [9]. Lo que es común en ellos es que la información histórica con la que trabajan está en el orden de los días, sin tomar en cuenta que la información de las redes fluye a grandes velocidades.

Desde hace algunos años, cuando las bolsas de valores hicieron posible que todas las transacciones se hagan de forma electrónica, se dejó el camino libre para que en el futuro no seamos los humanos quienes tomemos la decisión de comprar o vender acciones, sino sistemas computacionales avanzados que ofrecen tiempos de respuesta de alrededor de los segundos o pocos minutos, en los que debe ser factible hacer predicciones para tomar las mejores decisiones [2].

Para que sean efectivos los modelos predictivos en las bolsas de valores se debe trabajar con la información en tiempo real y aplicar técnicas de Big Data, debido a que los datos con los que se trabaja cumplen con sus características: velocidad, variedad y de gran volumen. El presente proyecto no trata con Big Data como tal, pero si usa sus herramientas.

2.1. Data mining

Data mining es el proceso de descubrir patrones en cantidades sustanciales de datos de forma automática o semiautomática. Los patrones encontrados deben ser significativos de modo que provean algún beneficio, como lo son las predicciones no triviales con nueva información. Estos patrones pueden presentarse de una forma estructural que puede ser estudiada y analizada, o simplemente como una caja negra [10]. El data mining está relacionado con el proceso de descubrir conocimiento en bases de datos, lo que actualmente y con el advenimiento del Big Data ha evolucionado en lo que se conoce como Data Science, que tiende a trabajar con información en tiempo real de distintas fuentes y diferente naturaleza.

El análisis de datos es un proceso que involucra los cinco pasos descritos en la Figura 1.

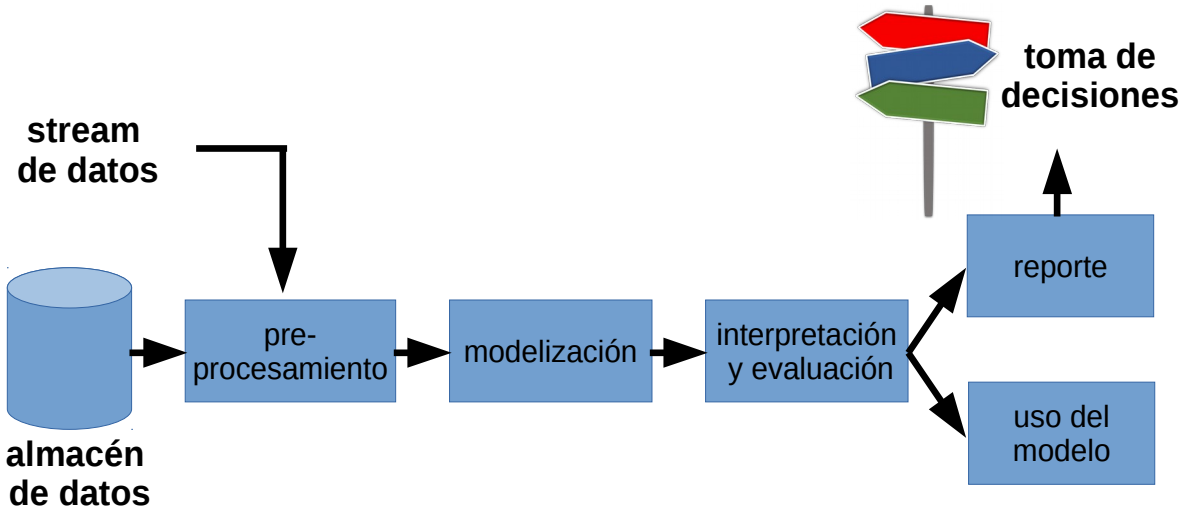

Figura 1: Proceso de minería de datos

2.1.1. Obtención de datos

Para analizar datos el primer paso es obtenerlos y estos pueden ser históricos por lotes (batch) o en tiempo real (stream). Generalmente se parte de datos históricos que pueden ser o no incrementados con datos en stream.

2.1.1.1. Datos en batch

Los datos en batch corresponden a los que están almacenados en alguna fuente y pueden ser de tres tipos:

1. **Estructurados:** Bases de datos, hojas electrónicas o en formato CSV.

2. **Semi estructurados:** Archivos en formato JSON o XML.

3. **No estructurados:** Texto plano como noticias o comentarios.

El concepto de almacén de datos ha sido concebido para replicar la información de bases de datos operacionales de una forma incremental y sin necesidad de salvaguardar toda la integridad referencial y propiedades que el modelo relacional exige. Generalmente un almacén de datos puede guardar información de cualquiera de los tres tipos.

2.1.1.2. Datos en stream

Los datos en stream son los que fluyen mientras transcurre el tiempo, y lo pueden hacer de una forma periódica o no. Estos datos pueden obtenerse de fuentes de diferente naturaleza, por ejemplo: sensores, redes sociales, logs de aplicaciones, etc. Los datos en stream se pueden procesar de dos formas:

1. **Realtime:** Cuando cada dato que arriba es procesado inmediatamente. Se lo usa en sistemas en donde el tiempo de procesamiento es mínimo y se necesita una respuesta inmediata. Ejemplo: un sistema antifraude de tarjetas de crédito.

2. **Near realtime:** Cuando periódicamente se procesan los datos que arriban en determinado intervalo de tiempo (segundos o pocos minutos). Ejemplo: un sistema de predicción del precio del Euro en USD cada cinco minutos.

2.1.2. Preprocesamiento de datos

El preprocesamiento de datos es la parte menos gratificante y que más tiempo consume del proceso de análisis de datos, debido a que estos pueden contener valores omisos o necesitar de varias transformaciones antes de que sean minados. Una de las tareas de preprocesamiento involucra la examinación del grado de correlación entre pares de datos numéricos [11]. Dependiendo de lo que represente cada atributo los datos omisos o ‘missings’ pueden ser tratados de diversa forma, aunque es común que sean llenados con un valor por defecto o que es calculado a partir de los demás datos. En esta etapa se analiza también si existen valores aislados o no balanceados que pueden ser causa de errores en la obtención de los datos, ya sean humanos o de los equipos encargados de esta tarea.

En el preprocesamiento de datos en stream no siempre se puede aplicar las técnicas usadas con datos en batch [12], por ejemplo cuando se llena un dato omiso con la media entre sus dos valores más próximos, con datos en stream aún no se tiene el siguiente dato y por lo tanto esto no se puede aplicar.

Cuando se obtienen datos semi estructurados o no estructurados es en esta etapa que se los transforma y se les da una estructura acorde a las posteriores técnicas de minería que se vayan a aplicar. El sentiment analysis entra en esta fase como una de esas herramientas, debido a que extrae del texto información como una valoración del sentimiento o la opinión.

2.1.3. Modelización de datos

En la modelización de datos se aplican las diferentes técnicas de minería para encontrar los patrones de los que se habló anteriormente. Mayormente se usan herramientas para modelización predictiva como el machine learning. Dentro de las estrategias para modelización de datos distinguen dos tipos: supervisadas y no supervisadas.

1. **Aprendizaje supervisado:** Involucra variables de entrada o independientes y variables de salida o dependientes (generalmente solo una). A las primeras también se les conoce como atributos y a las últimas como respuesta. En otras palabras, se tiene muestras de atributos y respuesta sobre las cuales se intentará entrenar una función que posteriormente prediga la respuesta sobre nuevas muestras de solamente atributos. Ejemplo: clasificación, regresión.

2. **Aprendizaje no supervisado:** Solamente se conocen los atributos y se deja a los algoritmos de aprendizaje que busquen posibles patrones previamente desconocidos en los datos. Ejemplo: clustering.

En las estrategias de aprendizaje supervisado, y específicamente cuando se trabaja con modelos estadísticos, se suelen distinguir como X a los atributos y como Y a la respuesta. El objetivo del proceso de modelización es encontrar una función f de forma que:

$$Y = f(X) + e$$

El valor ‘e’ corresponde a un error aleatorio con media cero que es independiente de X. La fórmula representa la información sistemática que X provee de Y [13]. El objetivo en esta etapa es encontrar esa función, a la que generalmente se le conoce como modelo.

La diferencia entre una clasificación y una regresión radica en la naturaleza de la variable de salida.

1. **Clasificación:** Respuesta cualitativa (categórica). Algunos métodos usados: regresión logística, árboles de decisión, naive Bayes, vecinos más cercanos, etc.

2. **Regresión:** Respuesta cuantitativa (numérica). También se le conoce como regresión. Algunos métodos usados: regresión lineal, árboles de decisión, regresión lineal generalizada (Gaussiana, Binomial, Poisson, Gamma), etc.

2.1.3.1. Regresión logística

La regresión logística es un método de aprendizaje estadístico que asigna a cada observación la probabilidad de pertenecer o no a una clase. Aunque puede trabajar con más de dos clases lo más común es que use para clasificaciones binarias. Para lograr que la variable de salida corresponda a una probabilidad debe limitarse a valores entre cero y uno, debido a eso se usa la función logística y el modelo queda así:

$$f(X)=\frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\beta_3 X_3 \ldots+\beta_n X_n}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\beta_3 X_3 \ldots+\beta_n X_n}}$$

La función f(X) devolverá la probabilidad, mientras más se acerque a cero o a uno pertenecerá a una u otra clase respectivamente. El objetivo es entonces estimar los coeficientes β_0, β_1, β_2, β_3,... β_n. El primero define la probabilidad por defecto cuando no se tiene valores de X. Los demás coeficientes corresponden a cada uno de los atributos que se hayan escogido para trabajar el modelo predictivo.

2.1.3.2. Linear Discriminant Analysis

Linear Discriminant Analysis o LDA, es método estadístico que se usa como un clasificador lineal o para reducir la dimensión de un dataset, ya que separa o relaciona dos o más variables. Lo que hace LDA es asumir que las variables involucradas siguen una distribución normal y además comparten la misma varianza. Partiendo de la función de densidad de probabilidad de cada variable y haciendo reducciones matemáticas llega a la ecuación matricial siguiente:

$$\delta_k(X)=X^T \Sigma^{-1} \mu_k-\frac{1}{2}\mu_k^T \Sigma^{-1} \mu_k+\log \eta_k$$

En donde **k** es el número de clases de la variable de respuesta, **X** es el vector de atributos para el cual se calculará dicha respuesta, **Σ** es una matriz de covarianza que es común a todas las clases, $\boldsymbol{\eta_k}$ son las probabilidades a priori de que un nuevo registro pertenezca a determinada clase y $\boldsymbol{\mu_k}$ es un vector de medias específico por clase. Los dos últimos valores se obtienen generalmente del dataset de entrenamiento. El nuevo registro **X** pertenecerá a la clase **k** cuyo valor de la función anterior sea el mayor.

2.1.3.3. Support Vector Machines

Support Vector Machines o SVM es también un método de aprendizaje supervisado que se puede usar como clasificador lineal, sin embargo no tiene una base estadística como los dos anteriores. El objetivo de SVM es encontrar un hiperplano que separe en el espacio cada uno de los puntos de la

muestra, a cada nuevo punto se asignará a la clase que corresponda según esta división en el espacio. Por hiperplano se entiende a un plano de dimensión n -1, siendo n el número de atributos del dataset, en otras palabras, si se trabaja en dos dimensiones entonces el hiperplano será la recta que divida a ambas clases, el objetivo de SVM es encontrar el hiperplano que mejor lo haga.

2.1.4. Interpretación y evaluación de resultados

La interpretación y evaluación de resultados implica determinar si la información obtenida en la etapa de modelización es relevante o no según los objetivos del análisis de datos. En caso de no ser relevante se debe volver a iniciar el proceso, escoger otros atributos, aplicar otros criterios de preprocesamiento o utilizar otros métodos de modelización. A continuación se describen algunas consideraciones a tener en cuenta cuando los resultados no han sido satisfactorios [11]:

1. **Datos de entrenamiento:** Pueden no representar de forma adecuada a la población debido a que no han sido seleccionados de forma aleatoria o su número es demasiado pequeño como para poder hacer generalizaciones a partir de ellos. Puede suceder también que tienen demasiados valores atípicos que entorpecen el proceso de modelización, o que en la etapa de preprocesamiento no se trataron correctamente.

2. **Datos de prueba:** Estos datos sirven para probar el modelo con información nueva que no fue utilizada en la etapa de entrenamiento. Deben ser escogidos también de forma aleatoria y en una proporción dependiendo de caso, es muy común usar un 80% de la muestra para el entrenamiento y un 20% para la evaluación. Una técnica bastante usada es la de Cross Validation [13], que divide la muestra en grupos iguales de observaciones de forma que uno de ellos sea el de prueba. Varias rondas de entrenamiento y evaluación se llevan a cabo de forma que todos los grupos de datos se hayan utilizado en ambos casos.

3. **Método de modelización:** Dependiendo de lo que representen los datos es probable que una técnica ofrezca mejores resultados que otra, por eso es importante probar algunas de ellas. Seguramente intentar con todas las que la herramienta informática disponga y que adecúen a la naturaleza de los datos será la mejor opción.

4. **Atributos:** La correcta selección, extracción y transformación de atributos puede resultar en un modelo más eficiente. Conocer detalladamente el dominio sobre el que se trabaja permitirá tomar las mejores decisiones. Es común normalizar los datos para que el modelo no se afecte por las diferencias cuantitativas en las unidades de medición de cada atributo.

5. **Parámetros:** Algunas técnicas de modelización tienen parámetros que se pueden ir ajustando hasta obtener mejores resultados.

Generalmente, cada método de aprendizaje tiene sus propias métricas de evaluación, aunque dependiendo de lo que representen los datos y la variable de respuesta se puede escoger una métrica que mejor se adapte a la realidad. Para clasificadores binarios es muy común recurrir a la matriz de confusión de la Tabla 1, en donde se comparan los valores que se predicen con los reales. Se entiende TP por verdadero positivo y FP por falso positivo, de forma similar para los valores negativos.

		predicción	
		P	N
real	**P**	TP	FN
	N	FP	TN

Tabla 1: Matriz de confusión binaria

En sistemas predictores interesa saber si la predicción fue hecha bien o no y la métrica sería el número de predicciones válidas sobre el total de predicciones, a esto se le conoce como accuracy o exactitud, que se formula de la siguiente forma:

$$Exactitud = \frac{TP + TN}{TP + FP + TN + FN}$$

2.1.5. Reporte y aplicación

Una vez que se ha seleccionado el modelo se lo puede aplicar a nueva información y hacer predicciones. Es importante evaluar continuamente la efectividad del mismo para hacer a tiempo los cambios que sean pertinentes, sobretodo en el caso de modelos que actualizan continuamente con cada nueva observación.

2.2. Sentiment Analysis

Actualmente se habla mucho del social media analytics que es el proceso de extracción de información útil a partir de los datos semi estructurados y no estructurados de las redes sociales: Facebook, Youtube, Instagram, Twitter, etc. Siete capas definen los diferentes tipos de datos que se pueden obtener: texto (tweets, comentarios), redes (contactos, seguidores), acciones (likes, visualizaciones), enlaces, localización, datos de móviles, y datos de buscadores. Además, se trata de un proceso iterativo de seis pasos: identificación, extracción, limpieza, análisis, visualización e interpretación [14].

Una de las áreas del social media analytics es el sentiment analysis, también conocido como opinión mining, que es un campo de estudio que analiza la opinión de las personas, sus sentimientos, evaluaciones, actitudes y emociones hacia otras entidades (personas, productos, servicios, eventos, compañías, etc.) y sus atributos [15]. Aunque es muy común hablar de sentimiento u opinión de forma indistinta, se debe tomar en cuenta que la opinión es la que lleva inmerso un sentimiento. Generalmente se intenta determinar si esta opinión es positiva o negativa, o dependiendo del caso, neutral.

Existen dos tipos de opiniones: directa y comparativa. La directa se forma de cinco partes [16]:

1. **Objetivo:** Es la entidad sobre la cual se genera la opinión.
2. **Atributo:** En caso de que la opinión no sea directamente sobre la entidad se especifica su atributo.
3. **Sentimiento:** Es la orientación de la opinión (positiva, neutral o negativa). Como se puede apreciar, se trata de un parámetro subjetivo.
4. **Poseedor:** Es el sujeto que genera la opinión.
5. **Tiempo:** Es el instante de tiempo en el cual se genera la opinión.

Por ejemplo en el siguiente tweet:

> El 2017/06/01 @joseluis escribió:
> “Las acciones de Apple están subiendo, es oportunidad de comprar o no vender”,

El objetivo es Apple, el atributo es el precio de sus acciones, el sentimiento es positivo porque incentiva a tomar una acción, el poseedor es José Luis (@joseluis) y el tiempo es 2017/06/01.

El sentimiento no necesita ser siempre categórico, se puede trabajar con un rango desde [-1, 1] que va desde negativo, pasa por neutral (0) y llega hasta positivo. De esta forma a cada opinión no solo se le da una polarización sino un valor de intensidad de esa polarización. De una forma más explícita, el sentimiento puede tener: orientación, intensidad o en algunos casos clasificación, como por ejemplo las cinco estrellas de las revisiones de los productos de Amazon [17]. El otro tipo de opinión es la comparativa, que evalúa algún atributo en dos entidades distintas.

El sentiment analysis puede hacerse sobre una oración, sobre un párrafo o sobre un documento. La complicación aumenta en ese mismo orden debido ya que la información puede presentarse desordenada de forma que dificulte su evaluación. La asunción que se tendrá en adelante es que cada oración expresa una sola opinión, sobre una sola entidad o atributo de la misma, lo que ciertamente no siempre se cumple.

2.2.1. Consideraciones

Puede resultar sencillo para una persona identificar cada uno de los componentes de una opinión, sin embargo, cuando se trata de un sistema computacional existen ciertas consideraciones que complican esta tarea y que son propias del lenguaje natural de las personas. Principalmente se identifica lo siguiente:

1. **Sinónimos y polisemia:** Dependiendo de la región, personas que hablan un mismo idioma pueden referirse a la misma entidad o atributo con distintas palabras (sinónimos) o pueden usar las mismas palabras para referirse a diferentes entidades (polisemia). Esto hace que el sentiment analysis deba adaptar sus diccionarios dependiendo de las características de los poseedores de las opiniones. De la misma forma cuando se intenta pasar de un dominio a otro.

2. **Sarcasmo:** Cuando la opinión es totalmente contraria a lo que literalmente se expresa, lo cual incluso suele ser difícil identificar para las personas.

3. **Oraciones compuestas:** Pueden ser fácilmente separables si llevan una conjunción o un signo de puntuación, pero en ocasiones pueden mezclar la información y hacer muy complicada la tarea.

4. **Subjetividad:** Una opinión que indica cierta característica propia de una entidad se la conoce como objetiva: “Las acciones de Microsoft bajaron su precio”. Por otra parte, una opinión que denota un sentimiento hacia ella desde el punto de vista del poseedor es subjetiva: “Espero que las acciones de Microsoft bajen su precio”. A la primera opinión se le puede otorgar mayor peso que la segunda, dependiendo del contexto.

5. **Negaciones:** Las negaciones en una oración cambian el sentido de la misma. Aparentemente es una tarea sencilla buscar los negadores (no, pero, es falso que, etc.) sin embargo, su distribución en el texto puede ser demasiado variable.

6. **Lenguaje social:** Desde que fue común el envío de mensajes cortos de texto (SMS) con una longitud limitada a 140 caracteres, las personas optamos por escribir de forma abreviada, ya sea suprimiendo caracteres o cambiándolos por símbolos (emoticones por ejemplo). A pesar de que en las redes sociales esta limitación ya no existe, salvo Twitter, esta costumbre se la sigue manteniendo. A esto se suman las faltas ortográficas o errores en la escritura que se incrementan por la informalidad que se maneja en estos medios.

2.2.2. Métodos

De forma general, los métodos usados en sentiment analysis son dos: los que se basan en el uso de vocabularios solamente, o los que usan vocabularios y además machine learning [18]. Los vocabularios o léxicos deben ser específicos por dominio y contienen solamente las palabras relevantes, es decir las que aportan información. Para el análisis de polaridad es conveniente separar el vocabulario en dos, palabras positivas y palabras negativas. Un tercer vocabulario puede ser el de las palabras que no aportan información como son los artículos y conjunciones, pero esto dependerá del método utilizado para obtener la polarización o intensidad de la opinión.

Los vocabularios no necesitan ser de una sola palabra debido a que las combinaciones de dos o más pueden cambiar su significado. Para esto se hace uso del término n-gram, en donde n corresponde al número de palabras sobre las que se basa el diccionario. Por ejemplo, en un vocabulario 1-gram la palabra “bien” puede calificarse como positiva, en cambio en un 2-gram la misma palabra puede calificarse como negativa si se le antepone un negador “no bien”. El primer paso de un proceso de sentiment analysis es la creación del vocabulario para el dominio en cuestión.

El sentiment analysis utiliza por lo general aprendizaje supervisado y el mayor problema que se encuentra es la falta de datasets de entrenamiento, es decir, que estén ya etiquetados como positivos, neutrales o negativos, o en su defecto, que tengan su valor de intensidad. Lograr un dataset de entrenamiento de forma manual es una tarea que consume demasiado tiempo. No sucede así con datasets de evaluaciones de productos como los de Amazon por ejemplo, en donde el mismo usuario ha clasificado la opinión en una escala de uno a cinco, sin embargo, no queda por demás revisar esta información y validar cuanto sea posible si el comentario corresponde con la clasificación.

Para usar una técnica de machine learning se debe obtener los atributos a partir del texto de cada oración, y una forma de lograrlo es eliminando primero las palabras o símbolos que no aportan información, para posteriormente aplicar una herramienta de POST (Part of Speech Tagging) que clasifica cada palabra restante como sustantivo, verbo, artículo, y demás, dependiendo tanto de su definición como del contexto de la oración. Posterior a esto se crean las combinaciones posibles de dos o más palabras dependiendo de los vocabularios que previamente se hayan obtenido.

Cuando se trabaja solamente con vocabularios se hace necesario aplicar reglas, que incluso pueden incluir ciertos símbolos, por ejemplo: un emoticón con una cara feliz en la oración, puede aportar información concluyente de que es positiva, o también, los signos de interrogación o admiración pueden incrementar en cierto valor la intensidad de la oración, independientemente si esta ya es positiva o negativa.

2.2.3. Paquetes de software y librerías

Considerando que el sentiment analysis conlleva procesamiento de lenguaje natural, existen diversas herramientas de propósito general como Freeling [19] y OpenNLP [20], que a modo de librerías ofrecen diferentes métodos para ejecutar las tareas que son comunes en este ámbito como: toquenización, segmentación de oraciones, etiquetado POST, extracción de entidades con nombre, etc. Con estas herramientas se puede trabajar desde cero y crear vocabularios acorde al tema que se necesite.

Un estudio reciente [21] de las herramientas actuales para procesamiento natural de lenguaje a nivel oración, sobretodo de redes sociales, demuestra que VADER (Valence Aware Dictionary and sEntiment Reasoner) ofrece buenos resultados trabajando especialmente con mensajes de Twitter, y es muy útil en casos en donde previamente no se tiene un vocabulario o datasets de entrenamiento. Este algoritmo ha sido utilizado con buenos resultados en [22] y [23], y validado como uno de los mejores de libre acceso actualmente en [18].

3. CONCEPTOS Y HERRAMIENTAS

En este capítulo se describe de forma general los conceptos y herramientas utilizadas en el desarrollo del presente trabajo, se limitará a resumir y dar una visión global de cada uno de ellos, enmarcada en los objetivos y alcance del proyecto. No se pretende justificar o dar una explicación exhaustiva de la teoría que respalda a cada concepto, sin embargo, se hará referencia a la bibliografía que ha servido de base para el desarrollo de cada apartado y que será de gran utilidad si se desea profundizar en los mismos.

Como herramientas se hace mención a lenguajes de programación, aplicaciones y librerías de software que han sido utilizadas. La información que se presenta de cada una ha sido, en su mayoría, extraída desde su documentación oficial.

3.1. Bolsa de valores

Una bolsa de valores es un mercado en donde, entre otros productos financieros, se comercializan acciones de compañías que se encuentran registradas en ella. Una bolsa es una entidad con fines de lucro, que funciona en un lugar determinado y ofrece los medios para garantizar que todas las transacciones se hagan de forma legítima. La función principal de una bolsa es permitir que las compañías obtengan capital de los inversionistas, y que estos a su vez, se beneficien de la rentabilidad de la compañía. Por otra parte están los intermediarios, que como profesionales especializados, obtienen beneficios al asesorar y encargarse de ejecutar las transacciones entre las compañías y los inversionistas a través de la bolsa. Entonces, se distinguen tres tipos de participantes principales en una bolsa de valores:

1. **Demandantes de capital:** Compañías, estados u organismos públicos.
2. **Oferentes de capital:** Inversores que compran acciones de las compañías.
3. **Intermediarios:** Corredores, agencias de valores, etc.

Existen diversas bolsas de valores alrededor del mundo y su importancia se define por el número y valor de las compañías que tienen registradas. En la Tabla 2 se muestran las seis más importantes a

nivel mundial según su capitalización expresada en millones de USD, a mayo de 2017, con valores extraídos de [24].

nombre	país	ciudad	# compañías	capitalización
New York Stock Exchange	Estados Unidos	Nueva York	2 304	20 388 427.4
NASDAQ	Estados Unidos	Nueva York	2 900	8 827 942.2
Japan Exchange Group Inc	Japón	Tokyo	3 557	5 424 014.0
Shanghai Stock Exchange	China	Shanghai	1 284	4 361 154.6
Euronext	Países Bajos	Amsterdam	1 281	4 059 354.8
London Stock Exchange Group	Reino Unido, Italia	Londres	2 489	4 047 981.2

Tabla 2: Principales bolsas de valores a nivel mundial

3.1.1. Acciones

Las acciones representan un derecho sobre los activos y ganancias de la compañía, a medida que se adquiera más acciones la participación en la propiedad de dicha compañía se hace mayor [25]. Cada acción es una proporción del patrimonio de la compañía y su valor es expresado en la moneda oficial de la bolsa de valores en donde cotice. Cada compañía tiene asignado un símbolo para su identificación, generalmente de dos a cuatro caracteres. En la Tabla 3 se muestran seis compañías con su símbolo, la bolsa en la cual cotizan y el valor que tuvo cada acción en USD al final del día 31 de mayo de 2017 (información obtenida de Yahoo Finance [26]).

nombre	bolsa de valores	símbolo	acción
Oracle Corporation	NYSE	ORCL	45.39
Sony Corporation	NYSE	SNE	36.62
Johnson & Johnson	NYSE	JNJ	128.25
Facebook Inc	NASDAQ	FB	151.46
Apple Inc	NASDAQ	AAPL	152.76
Microsoft Corporation	NASDAQ	MSFT	69.84

Tabla 3: Acciones de compañías, símbolos y valores

El valor de una acción varía según la ley de la oferta y la demanda, pero esta se encuentra afectada por diversos factores entre los que destacan la especulación, las noticias y pronunciamientos sobre algún suceso respecto a la compañía. Con el surgimiento del Internet y las redes sociales esta información se

transmite a nivel mundial en segundos, lo que obliga a tomar decisiones de forma rápida. Por ejemplo en la Figura 2 se muestra la evolución las acciones de Apple Inc (AAPL) para en el año 2016.

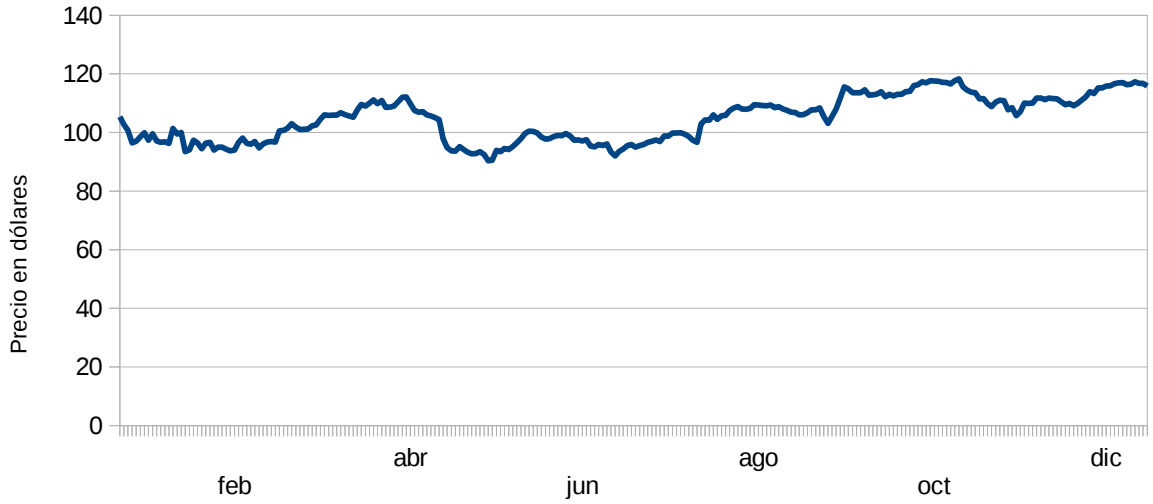

Figura 2: Evolución del precio de las acciones de Apple Inc en el 2016

3.1.2. Índices bursátiles

Un índice bursátil es un indicador que se construye a partir de los valores de las acciones de un número determinado de compañías que comparten determinadas características, generalmente corresponden a las más importantes de una bolsa de valores o de un país. El número de compañías, las condiciones para que sean consideradas y la forma de cálculo son propias de cada índice. El objetivo de los índices bursátiles es dar una representación general de la evolución de determinado mercado. En la Tabla 4 se muestran varios de los más conocidos actualmente.

nombre	bolsa de valores
Promedio Industrial Dow Jones	30 mayores compañías que cotizan en Estados Unidos
S&P 500	500 mayores compañías que cotizan en el NYSE y NASDAQ
IBEX 35	35 mayores compañías que cotizan en España
Nikkei 225	225 mayores compañías que cotizan en Japón

Tabla 4: Índices bursátiles de ejemplo

3.1.3. Fondos de inversión cotizados en bolsa

Los fondos de inversión cotizados en bolsa (ETF, Exchange Traded Funds) son fondos de inversión que cotizan en determinada bolsa de valores, tienen un símbolo y sus acciones actúan como las de cualquier otra compañía, incluso repartiendo dividendos. Generalmente intentan seguir de cerca el

comportamiento de algún índice bursátil, ya sea comprando acciones de las compañías en las mismas proporciones que forman el índice, o solamente de las más representativas. En la Tabla 5 se muestran tres fondos de inversión que siguen el índice S&P 500.

nombre	símbolo
SPDR S&P 500 Trust ETF	NYSE: SPY
Vanguard S&P 500 ETF	NYSE: VOO
iShares S&P 500 Index	NYSE: IVV

Tabla 5: Fondos de inversión cotizados en bolsa que siguen el índice S&P 500

3.2. Twitter

Twitter es una red social para compartición mensajes en línea que pueden abarcar comentarios, estados de ánimo, noticias o ideas en general. Se usa a nivel mundial a través de su sitio web o dispositivos móviles, con alrededor de 313 millones de usuarios activos mensualmente que se comunican en más de 40 idiomas diferentes [27]. Sus oficinas principales están en San Francisco, Estados Unidos.

Los mensajes publicados en Twitter, conocidos como tweets, están limitados a 140 caracteres y es debido a que la red social fue concebida como un servicio en Internet de mensajes cortos de texto (SMS) similar al servicio ofrecido por las operadoras de telefonía móvil.

3.2.1. Usernames, hashtags y cashtags

Crear una cuenta en Twitter no tiene costo y este paso es necesario para poder publicar tweets, los que por defecto son visibles para todo el mundo, a menos que se lo restrinja a un grupo de usuarios. Cada usuario puede ser seguido y seguir a otros, lo que significa que al acceder a la red social se le presentará solamente los mensajes que publiquen estos últimos. Los usuarios no necesitan ser personas, pueden ser también compañías o áreas específicas de estas, o productos de una compañía, entre otros casos.

Cada usuario necesita un ‘username’ que escoge al crear su cuenta, el que además de servir para identificarse como usuario de la red, le permite ser referenciado en los tweets de otros usuarios, para lograr esto se antepone el símbolo ‘@’ al username. Además, para resaltar que el contenido de los tweets hace referencia a un tema determinado, se hace uso de los hashtags, que también son identificadores como los usernames, pero estos van precedidos del signo ‘#’.

A pesar de que Twitter no maneja preferencias para la asignación de usernames, es muy común que los usuarios lo hagan corresponder con su verdadero nombre (en caso de estar disponible). Por ejemplo @apple y @microsoft para Apple y Microsoft respectivamente. Los hashtags, por otra parte, no corresponden a un usuario en particular, y un username puede ser a la vez hashtag, por ejemplo #apple y #microsoft. Se acostumbra a usar hashtags hacer mención al usuario en el cuerpo de un tweet, mientras que los usernames permiten incluir al usuario como parte de la conversación, o en otras palabras, dirigir un mensaje a ese usuario. Cada tweet puede iniciar una conversación si más usuarios responden a ese mensaje con otros. También, un usuario puede compartir o reenviar un tweet publicado por otro usuario, lo que se conoce como 'retweet'.

Para diferenciar la información de cotización en bolsa de una compañía se hace uso de los 'cashtags' en lugar de los hashtags, y corresponden a los símbolos que las compañías usan en la bolsa de valores precedidos de '$', por ejemplo $AAPL y $MSFT.

Los usernames, hashtags y cashtags permiten hacer consultas específicas de tweets que correspondan a un usuario o un tema, o ambos a la vez. El contenido de un tweet aunque es puramente texto, también puede contener enlaces a otros sitios web. En la Figura 3 se muestra un tweet de ejemplo, el usuario que lo ha publicado es @andrejg_au y hace referencia a otro usuario @themotleyfool, que corresponde al del sitio web en donde está el artículo del enlace incluido en el tweet. También ha señalado los hashtags #stocks, #ai y #ux, para enfatizar que el contenido se refiere a acciones, inteligencia artificial y experiencia de usuario respectivamente. Y por último, ha incluido el cashtag $AAPL que corresponde a la compañía Apple.

Figura 3: Tweet con usernames, hashtags y cashtags

3.2.2. Tweets en batch y stream

Twitter permite a sus usuarios acceder a varias APIs (Application Programming Interface) [28] para obtener tweets bajo ciertos parámetros: usernames, hashtags, cashtags, fechas, localización, etc. La REST API permite acceder a tweets publicados en el pasado, mientras que con la Streaming API se

pueden obtener los que se van publicando al instante. La primera es limitada en cuanto al número de mensajes que se obtiene debido a que Twitter provee solamente los que considera más relevantes. La segunda en cambio, devuelve todos los mensajes disponibles pero es menos flexible en los filtros que se puede aplicar, además que es susceptible a caídas en la comunicación, que harían perder tweets.

Otra forma de extraer tweets es mediante software específico para este propósito, entre ellos se destaca twiQuery [29], el cual permite descargar tweets bajo parámetros similares a los de la API de Twitter, además, incorpora un filtro para detectar tweets con emoticones que demuestren un sentimiento positivo o negativo. Una tercera forma de acceder a tweets, y seguramente la menos eficiente, es mediante el archivo HTML que se obtiene al guardar la página después de hacer una consulta en un navegador directamente desde la web de Twitter, para posteriormente y mediante un script que busque las etiquetas correspondientes, extraer la información de los mensajes desde el código del archivo.

3.3. Python

“Python es un lenguaje de programación fácil de aprender y potente. Tiene estructuras de datos eficientes de alto nivel y un enfoque simple pero efectivo para la programación orientada a objetos. Su sintaxis elegante y su dinámica tipificación, junto con su naturaleza interpretada, lo convierten en un lenguaje ideal para el rápido desarrollo de scripts y aplicaciones en muchas áreas en la mayoría de las plataformas” [30].

El éxito de Python radica en que, al ser open source, contiene librerías que han sido aportadas por la comunidad de desarrolladores, enfocadas en diversas áreas de estudio. A continuación se describen dos que se usan en este trabajo.

3.3.1. Scikit Learn

Scikit Learn [31] es una librería para machine learning que incluye herramientas para minería y análisis de datos. Está basada en NumPy, SciPy y matplotlib, que son otras librerías de Python para análisis numérico, científico y trazado de gráficos respectivamente. La incorporación de esta librería a Python ha hecho que el lenguaje sea usado enormemente en tareas de data science, e incluso que junto al lenguaje para desarrollo estadístico R [32], sean los que más se usen en este campo [33].

La librería incluye métodos para aprendizaje supervisado y no supervisado, selección y evaluación de modelos, carga y transformaciones de datasets, etc. Al usar Python para análisis de datos se hace imprescindible utilizar también Jupyter [34], que es una interfaz de comandos vía web al estilo notebook, en donde se puede trabajar interactivamente con pipelines de machine learning.

3.3.2. Natural Language Toolkit

Natural Language Toolkit [35] o NLTK es una librería para procesamiento de lenguaje natural, específicamente para el idioma Inglés. Incorpora métodos para crear vocabularios, procesar texto sin formato, escribir programas estructurados, categorizar y etiquetar palabras (POST), clasificar y extraer información del texto, analizar la estructura de oraciones y construir diccionarios basados en atributos. Además, provee interfaces para diversos recursos vocabularios que tienen directa aplicación en procesos de sentiment analysis. NLTK se ha convertido en una de las principales herramientas en su área, debido a que es gratuita, de código abierto y relativamente sencilla de utilizar.

3.4. Apache Spark

Apache Spark es una plataforma para computación distribuida que puede funcionar en clusters de nodos del orden de los miles. Se introdujo como una alternativa de procesamiento a Hadoop MapReduce [36] que disminuye el tiempo de ejecución hasta en 100 veces [37]. Su éxito radica en que utiliza la memoria como primer medio de almacenamiento, lo que evita los altos retardos de acceso a disco. Sobre otras plataformas de computación distribuida Spark ofrece estas ventajas [38]:

- Una plataforma de ejecución más eficiente para machine learning y análisis de datos interactivo.
- Una sola pila de librerías para procesar datos en batch y en stream mediante lenguaje SQL, procesado de grafos y análisis de datos complejo.
- Provee un API de alto nivel para trabajar con computación distribuida sin necesidad de conocer los detalles de la misma.
- Soporte para diversos medios y formatos de almacenamiento: bases de datos relacionales, y no relacionales (HBase, Cassandra, Parquet, MongoDB, HDFS, Amazon S3)

La principal abstracción de Spark son los RDDs (Resilient Distributed Datasets) que son colecciones de datos tolerantes a fallos que pueden operar en paralelo, sin embargo, desde Spark 2.0 se incentiva al uso de DataFrames, colecciones similares a los RDDs que pueden ser accedidos y procesados al estilo SQL. Un DataFrame es como una tabla de una base de datos relacional, un Dataframe de la librería Pandas en Python, o un data.frame de R, con la diferencia que puede operarse en paralelo. Cuando se lee un dataset desde una fuente cualquiera de las que soporta Spark, este se convierte en un DataFrame.

Existen dos métodos para trabajar con DataFrames, las transformaciones y las acciones. Las primeras producen más DataFrames pero no son ejecutadas hasta que se llame a una acción. Las

transformaciones permiten hacer el preprocesamiento del dataset hasta que está listo para el proceso de análisis.

3.4.1. Librerías

Spark está escrito en el lenguaje de programación Scala, pero ofrece también APIs de alto nivel para Java, Python o R. Sobre su núcleo funcionan cuatro módulos en forma de librerías, los que se muestran en la Figura 4. Spark SQL es generalmente el punto de partida para cualquier aplicación, debido a que contiene funciones para acceder a información en distintos formatos, es una potente herramienta para llevar acabo procesos extracción y preprocesamiento de datos. GraphX por otra parte, está orientado a la programación distribuida de grafos. Spark Streaming y Mllib se describen con más detalle en las dos secciones siguientes.

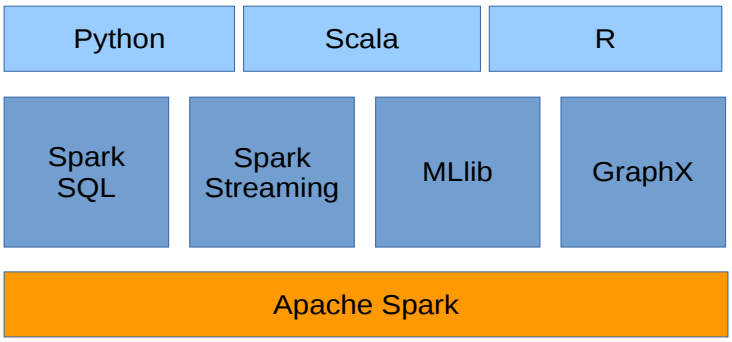

Figura 4: Componentes de Spark

3.4.1.1. Spark Streaming

Spark Streaming permite el procesamiento distribuido de grandes flujos de datos en tiempo real. La abstracción que utiliza se denomina Dstreams (Discretized streams), que son secuencias de RDDs que se extraen de diversas fuentes, como por ejemplo: mensajes de sensores (IoT), mensajes de Twitter, logs de sistema operativo, la lectura automática de archivos de texto, sockets TCP, etc.

Lo que hace Spark Streaming es dividir la información que llega en intervalos de tiempo configurables que pasan a ser procesados por el núcleo de Spark. En la Figura 5 se representa este concepto.

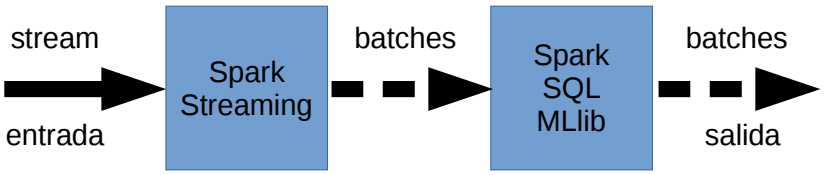

Figura 5: Funcionamiento de Spark Streaming [39]

3.4.1.2. Spark MLlib

Spark Mllib es la librería de machine learning de Spark que ofrece las siguientes funcionalidades [40]:

- Algoritmos de machine learning para clasificación, regresión, clustering y filtrado colaborativo.
- Extracción y transformación de atributos, reducción de dimensión, etc.
- Herramientas para construir, evaluar y poner a punto de pipelines de machine learning.
- Persistencia para la carga y el almacenamiento de algoritmos, modelos y pipelines.
- Utilidades de álgebra lineal, estadística, etc.

Una funcionalidad importante de Spark es que dispone de algoritmos de machine learning que pueden funcionar con datos en streaming. En la versión 2.1, que es la que se utilizará en este trabajo, se disponen de: regresión lineal, regresión logística (clasificación) y k-means (clustering). La ventaja de tener algoritmos de machine learning en tiempo real es que el modelo se va actualizando con la información que va llegando a Spark, sin necesidad de tener que calcular nuevamente el modelo.

3.4.2. Modo cluster

Spark puede funcionar solo, o mediante un administrador de cluster como lo son Hadoop YARN [36] o Apache Mesos [41], en cualquier caso, la estructura que sigue es la de la Figura 6.

El driver es el nodo desde el cual se inicia la ejecución de la aplicación (program), se comunica con el cluster manager para utilizar los recursos del cluster. Spark ofrece un intérprete de comandos similar al de Python, este es un tipo de driver program. El driver debe tener comunicación con los workers permanentemente debido a que comparte con ellos la información necesaria para la ejecución, además que recibe los resultados de cada acción que se vaya ejecutando. No se necesita que el driver se ejecute en un nodo independiente, puede hacerlo desde un worker o incluso, aunque no recomendable, desde el cluster manager. Dentro del driver program estará un objeto llamado SparkContext, que es

quien se comunica con el cluster manager para solicitar los recursos necesarios para iniciar la ejecución de la aplicación.

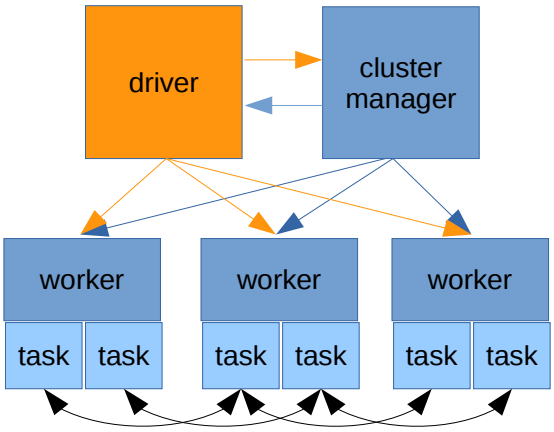

Figura 6: Arquitectura de cluster de Spark

Como se mencionó anteriormente, el cluster manager puede ser YARN, Mesos o incluso el mismo Spark. Los dos primeros están especializados en brindar esta funcionalidad y son los recomendables para ambientes en producción, el tercero en cambio funciona muy bien en ambientes de prueba y su implementación es muy rápida de llevar a cabo.

Los workers son realmente quienes hacen el trabajo duro, se encargan de ejecutar la aplicación obedeciendo al driver. Cada worker contiene una o más instancias llamadas executors que están asociados a una aplicación del driver, son estos quienes mediante la creación de tareas se encargan de llevar a cabo el trabajo. Cada worker debe tener acceso a los datos que necesita procesar, el driver no se hace cargo de compartirlos, lo recomendable es usar sistemas de ficheros en red.

Aunque Spark abstrae muchos detalles de la programación distribuida hay ciertos conceptos que deben estar claros antes de crear y ejecutar una aplicación, las variables compartidas es uno de estos. Como cada worker tiene una copia de las variables que se utilizan en la aplicación, no se garantiza que los cambios (lecturas y escrituras) se coordinen entre todos ellos, para esto Spark ofrece dos objetos: las broadcast variables que son de solo lectura y los accumulators que sirven como contadores que se sincronizan en todo el cluster.

4. METODOLOGÍA

En este capítulo se aborda la descripción del trabajo a realizar para evaluar el impacto que tiene incluir información de redes sociales en la predicción del precio de acciones, así como también las posibles mejoras de actualizar el modelo permanentemente con la nueva información que se va obteniendo en tiempo real. Se indica también la arquitectura y los recursos utilizados para el desarrollo de todo el trabajo.

4.1. Descripción

Tomando en cuenta la importancia en la bolsa de valores y la actividad en redes sociales que tienen compañías tecnológicas como Apple, Microsoft, Google, Amazon y Facebook, se ha escogido un índice que tenga estrecha relación con estas, y es el Standard & Poor's 500, conocido abreviadamente como S&P 500, que es calculado a partir de los precios de acciones de las 500 compañías más importantes entre el NYSE y el NASDAQ. En la Tabla 6 se detallan las diez primeras compañías en orden de importancia que forman el S&P 500 [42], en donde se incluyen las antes mencionadas. Estas diez compañías se tomarán en cuenta en este trabajo.

símbolo	**nombre**
AAPL	Apple Inc
MSFT	Microsoft Corporation
AMZN	Amazon.com Inc
FB	Facebook Inc A
JNJ	Johnson & Johnson
XOM	Exxon Mobil Corp
BRK.B	Berkshire Hathaway Inc B
JPM	JPMorgan Chase & Co
GOOGL	Alphabet Inc A
GOOG	Alphabet Inc C

Tabla 6: Compañías más representativas del índice S&P 500

Debido a limitaciones de disponibilidad de datos correspondientes a las fluctuaciones del índice S&P 500 en intervalos cortos de tiempo disponibles en Internet, se ha optado por utilizar un fondo de inversión cotizado en bolsa que sigue ese índice y es el SPDR S&P 500 ETF Trust, el cual se comercializa en el NYSE y cuyo símbolo es SPY.

Tomando en cuenta que un inversor tiene dos opciones respecto a una acción que posee: vender o no vender, u otras dos respecto a una acción que no posee: comprar o no comprar; más que predecir el precio de una acción resulta ventajoso saber si subirá o bajará su precio en el futuro [5]. Debido a esta razón no se intentará predecir el valor del precio de la acción después de un tiempo determinado sino la dirección de su variación, es decir si subirá o bajará su precio.

4.1.1. Datos utilizados

Son dos los datos que se han escogido para este trabajo: precios de acciones y tweets relacionados con esas acciones. Los precios de acciones porque es lo que se intentará predecir y los tweets debido a que es información que fluye rápidamente y es viable filtrarla de modo que solamente se obtenga la que hace referencia a las acciones de determinadas compañías.

Se trabajará con información de 14 semanas, específicamente desde el 9 de marzo hasta el 14 de junio de 2017, en total suman 98 días calendario.

4.1.2. Intervalos de tiempo

Como se ha mencionado en capítulos anteriores, por la falta de estudios relacionados que trabajen en intervalos cortos de tiempo, y además, debido a la naturaleza y disponibilidad de los datos necesarios para este proyecto, se ha elegido hacer predicciones de la variación del precio de acciones en tres casos de forma separada: cada 5, 15 y 30 minutos.

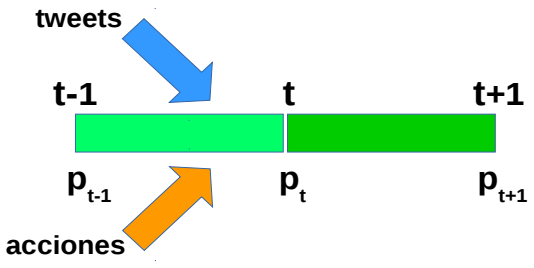

Figura 7: Captación de datos e instante de cálculo

En la Figura 7 se describe el intervalo de tiempo en el que se capta la información y el instante en el que se hace la predicción. Al final del intervalo (instante **t**) y con la información captada entre **t-1** y **t**,

se hace la predicción de la dirección que tendrá el precio de la acción al final del siguiente intervalo (instante **t+1**). En la descripción se ha tomado en cuenta que el flujo de tweets es permanente pero no periódico, mientras que el precio de las acciones es periódico y se lo obtiene una sola vez al final del intervalo.

4.2. Primera parte: Datos en batch

En esta primera parte se utiliza Python y sus librerías de machine learning para crear el modelo predictivo con los atributos que se extraen de la etapa de preprocesamiento. Se evalúan tres métodos de machine learning de los que proveen las mencionadas librerías. En la Figura 8 se muestra cada una de las etapas que involucra esta primera parte.

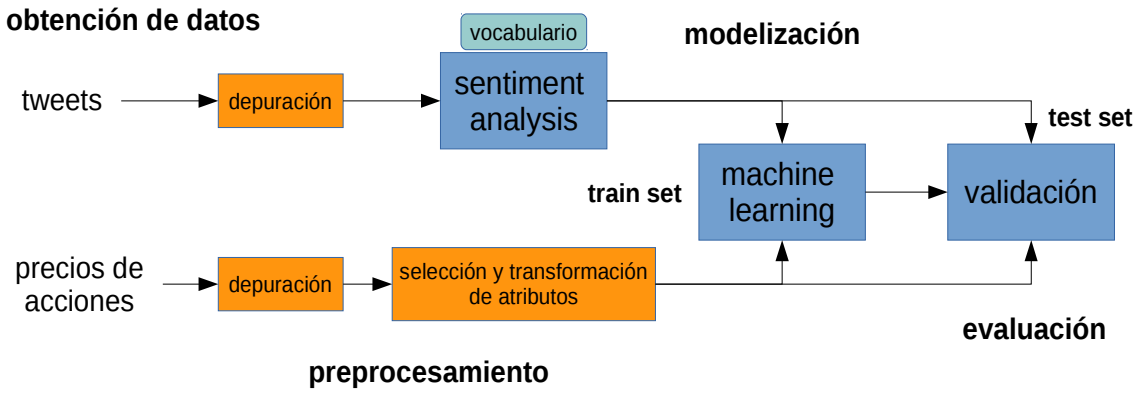

Figura 8: Esquema de la primera parte con datos en batch

La etapa de procesamiento se detalla en el siguiente capítulo, e incluye la extracción de atributos de los tweets mediante una técnica de sentiment analysis. Del dataset de 98 días con el que se trabajará se separará los últimos 21 días (3 semanas) para hacer la validación (test set). Se han escogido estas tres últimas semanas porque aproximadamente corresponden al 20% del dataset, que además servirá para comparar con los resultados de la segunda parte.

4.3. Segunda parte: Datos en stream

En la segunda parte se utilizará Spark y sus librerías de machine learning con datos en modo stream, la Figura 9 detalla el esquema de esta parte. Los datos son los mismos con los que se trabaja en modo batch, los registros de los primeros 77 días servirán para entrenar inicialmente el modelo y con los registros de los últimos 21 días este se irá actualizando, después de hacer la respectiva predicción en cada intervalo de tiempo, la que al final servirá para evaluar el resultado de aplicar esta arquitectura.

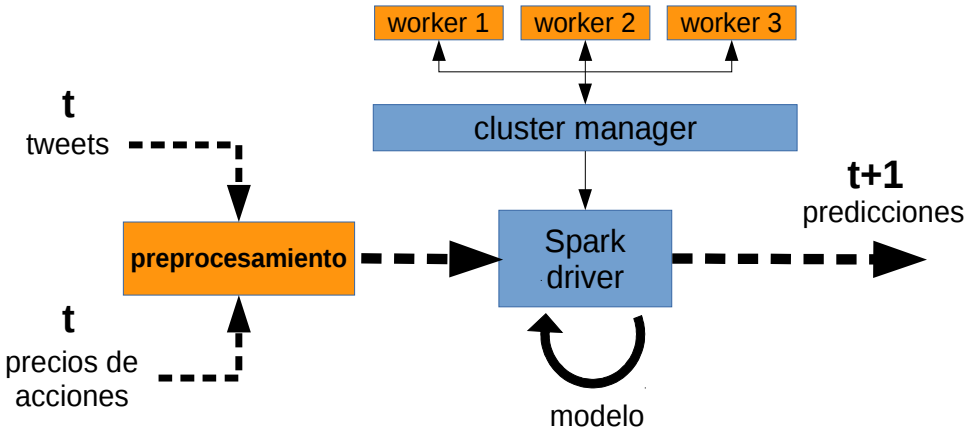

Figura 9: Esquema de la segunda parte con datos en stream

Se utilizara el mismo cluster manager de Spark con tres workers en un ambiente virtualizado con contenedores de Docker [43]. Los detalles de esta implementación se verán más adelante.

5. OBTENCIÓN Y PREPROCESAMIENTO DE DATOS

Como se mencionó en el capítulo previo, se trabajará con la información de precios de acciones de diez compañías (AAPL, MSFT, AMZN, FB, JNJ, XOM, BRK.B, JPM, GOOGL, GOOG) más el fondo de inversión SPDR S&P 500 ETF Trust (SPY), y los mensajes de Twitter que hacen mención a estas acciones. La información comprenderá desde el 9 de marzo hasta el 14 de junio de 2017.

5.1. Precios de acciones

Generalmente los precios de las acciones en tiempo real no están disponibles de forma gratuita, las bolsas de valores suelen comercializar esta información en forma de servicios, los que están orientados a intermediarios que realizan transacciones diariamente. Lo que está disponible en Internet es un histórico de los precios de las acciones con corte diario. En la Tabla 7 se observa una muestra para cinco días de los datos extraídos para el símbolo SPY de Yahoo Finance [26]. La información es similar para todas las demás compañías, sin embargo en los siguientes apartados se tomará como ejemplo solamente a SPY.

date	open	high	low	close	adj close	volume
2017-05-05	239.19	239.72	238.68	239.70	239.70	62001300
2017-05-08	239.75	239.92	239.17	239.66	239.66	48385700
2017-05-09	239.96	240.19	239.04	239.44	239.44	51363200
2017-05-10	239.39	239.87	239.15	239.87	239.87	54293800
2017-05-11	239.35	239.57	238.13	239.38	239.38	62358300

Tabla 7: Precios históricos de SPY

La descripción de cada una de las columnas es la siguiente:

- **Open:** Precio de apertura, corresponde al precio de la acción de la primera transacción en el día.
- **High:** Precio más alto al que se comercializó la acción en el día.
- **Low:** Precio más bajo al que se comercializó la acción en el día.

- **Close:** Precio de cierre, corresponde al precio de la acción de la última transacción en el día.
- **Adj Close:** Precio de cierre ajustado a la división de acciones (split) en el pasado. No se tomará en cuenta en este trabajo debido a que no han habido cambios de este tipo en las acciones en el periodo analizado.
- **Volume:** Número de acciones que se comercializaron en el día.

Es importante recalcar que salvo ciertas coincidencias, el precio de apertura no es igual que el de cierre del día siguiente. Además, no tiene mucho sentido hablar de información en tiempo real como la de la Tabla 6, debido a que se calcula en un periodo de tiempo específico. En cambio, lo que sí tiene sentido es un flujo constante de los precios y volumen de las acciones que se van comercializando al instante, a partir de la cual se podría crear registros similares para diferentes intervalos de tiempo.

5.1.1. Obtención y depuración de datos

Los datos se han obtenido del sitio web de Alpha Vantage [44], que los provee en intervalos de 1, 5, 15, 30 y 60 minutos. No se trata de datos en stream, sino de información que debe ser consultada a través de una solicitud HTTP en el intervalo deseado y con el URL correspondiente. La respuesta la entrega en formato JSON.

Para el intervalo de días planteados en este proyecto se ha logrado obtener solamente la información de las acciones cada 5 minutos, pero a partir de esta se ha creado los datasets con información cada 15 y 30 minutos. Los atributos Open y Close corresponden a los del primer y último registro respectivamente, en cambio para los atributos High y Low se calcula el máximo y mínimo en cada intervalo.

Previo a lo descrito en el párrafo anterior se ha verificado si había inconsistencia en los datos, tomando en cuenta que deben corresponder solamente a los días que la bolsa de valores estuvo abierta, debido que aparte de sábados y domingos existieron dos días festivos (14 de abril y 29 de mayo). Además, se verificó que para cada día existiesen 79 registros (09:30 a 16:00 cada 5 minutos) y se encontró que había registros con distintos precios y en la misma hora para el día 1 de junio, lo que se entendió como un error de Alpha Vantage y se decidió tomar solamente los últimos valores aportados, ya que los iniciales y finales del día se comprobaron con los que Yahoo Finance proporciona para ese mismo día.

Además, existieron ciertos missings para algunas acciones en días distintos. El criterio que se ha utilizado para llenar estos valores es el conocido en Python como “ffill” o forward fill, que es llenar los missings con el último valor conocido de cada atributo.

Para tener una idea de cómo varía el precio de las acciones en la Figura 10 se representa el precio de la acción de SPY en el periodo del 09 al 30 de marzo de 2017. Los datos han sido tomados cada treinta minutos.

Figura 10: Precio de la acción de SPY del 9 al 30 de marzo de 2017 cada 30 minutos

5.1.2. Selección y transformación de atributos

Inicialmente se podría trabajar con Open, High, Low, Close y Volume como atributos, pero se ha optado por seguir el trabajo realizado en [45] en el que aplica una reducción de dimensión. Se extraen dos nuevos atributos que muestran el cambio porcentual del precio de la acción en determinado intervalo y son las siguientes:

- $High\,Low\,Percentage\,(HLP) = \frac{high - low}{low}$
- $Close\,Open\,Percentage\,(COP) = \frac{close - open}{close}$

Además, y antes de concluir el preprocesamiento de este dataset se calcula la dirección de la variación del precio de la acción en el siguiente intervalo de tiempo, que será la etiqueta o clase para posteriormente entrenar el modelo. Considerando que un poseedor de acciones no pierde si el precio se mantiene, el criterio que se ha utilizado se indica a continuación:

- $Direction = \begin{cases} Up\ si\,(close\,(t+1) - close\,(t) \geq 0) \\ Down\ si\,(close\,(t+1) - close\,(t) < 0) \end{cases}$

Se ha creado un dataset para cada acción y en cada intervalo de tiempo de los tres que se van a analizar. En la Tabla 8 se muestra como queda parte del dataset de SPY para el intervalo de 5 minutos.

datetime	close	volume	hlp	cop	direction
2017-03-09 09:30:00	236.72	233286	0.000127	0.000042	Down
2017-03-09 09:35:00	236.54	455561	0.001142	-0.000803	Up
2017-03-09 09:40:00	236.73	167082	0.001015	0.000803	Up
2017-03-09 09:45:00	236.81	196563	0.000507	0.000296	Up
2017-03-09 09:50:00	236.92	271478	0.001056	0.000422	Up

Tabla 8: Estructura del dataset de acciones de SPY

5.1.3. Descripción de la aplicación para extraer acciones en realtime

Como ya se ha mencionado, lo que se necesita hacer es extraer las acciones cada cierto intervalo de tiempo mediante una consulta HTTP al sitio web de Alpha Vantage. Para esto se ha creado un script en Python que periódicamente hace la solicitud, procesa el texto en formato JSON y va acumulando los valores separados por comas en un archivo de texto. Hay que tomar en cuenta que se puede hacer una solicitud compacta y una completa, en la primera se devuelven solamente los diez últimos registros, en cambio en la completa se devuelven los del último mes aproximadamente. Esta ventaja hace que en problemas de conexión cuando se han perdido registros se pueda volver a obtenerlos. Sin embargo, si este módulo va a ser parte de una aplicación en tiempo real es necesario considerar un tiempo de espera limitado a la respuesta del sitio web, para evitar problemas de retardo. Más adelante se verá que con Spark no hace falta la inclusión de otros módulos debido a que todo se programa en la misma aplicación.

5.2. Mensajes de Twitter

La información que se obtuvo de Twitter son los mensajes que se publican con los cashtags correspondientes a los once símbolos de las acciones que se va a utilizar: $AAPL, $MSFT, $AMZN, $JNJ, $FB, $XOM, $BRK.B, $JPM, $GOOGL, $GOOG, $SPY. Cada mensaje puede contener uno o varios de ellos y se ha escogido solamente los que están escritos en idioma Inglés.

5.2.1. Obtención y depuración de tweets

Para este trabajo se ha optado por extraer la información desde la API realtime de Twitter, sin embargo, también se ha extraído desde archivos HTML y twiQuery para ciertos días al inicio del

intervalo y también cuando existió algún problema con la conexión, procurando obtener todos los mensajes que hayan hecho referencia a los cashtags en cuestión, incluyendo sábados, domingos y días festivos.

Cada tweet contiene además de su fecha y hora de publicación, información adicional como el usuario que lo publicó, coordenadas desde donde se publicó, un identificador único para cada mensaje (que sirve para eliminar duplicados), entre otros. Sin embargo, y siguiendo los objetivos de este trabajo se almacenó solamente la fecha, hora y el mensaje. En la Tabla 9 se muestra cuatro tweets de ejemplo extraídos del dataset.

datetime	tweet
2017-05-30 06:25:00	Looks like S&P 500 is on course to 2440-2455 before correcting (2455 has been my call since Nov 9.) $SPY #POTUSpic.twitter.com/I16t1oH0h5
2017-05-30 07:17:00	Just an $FYI $AMZN $FB $AAPL $MSFT and $GOOGL are responsible for more than 40% of the S&P 500 return YTD #markets #stocks #investing
2017-05-30 08:46:00	$AMZN hit 99.99 in premarket earlier.

Tabla 9: Dataset con tweets en texto sin procesar

Es importante mencionar que la fecha y hora que devuelve la API de Twitter para cada mensaje corresponde al lugar de la consulta, por lo que se lo ajustó al de la zona horaria del dataset de la bolsa de valores. El NYSE está en Nueva York, con zona horaria UTC-5 mientras que Barcelona se encuentra a UTC+1 (una hora más en cada uno en horario de verano), por lo que se restó seis horas a la fecha antes de almacenar la información y seguir con el siguiente paso.

Los tweets son solamente texto y contienen datos que no aportan información como son enlaces a imágenes o noticias que están en el mismo Twitter o en otros sitios web. Lo que se ha hecho es eliminar esta información para que el texto quede limpio, los pasos que se han seguido se detallan a continuación:

1. **Eliminar los enlaces a imágenes:** Se ha eliminado toda palabra que empiece con “pic.twitter.com”.
2. **Eliminar enlaces a otros sitios web:** Todo el texto que empieza con “http://” y “https://”
3. **Eliminar usernames y hashtags:** Toda palabra que empieza con “@” y “#”
4. **Eliminar símbolos:** Se ha encontrado “>” y “<” en algunos tweets.
5. **Eliminar espacios en blanco:** Los que quedan al inicio y final de cada tweet

Los cashtags solamente sirven para asociar los tweets con determinada acción. Entonces, una vez que se ha filtrado y separado un dataset para cada compañía, se ha eliminado también los cashtags. Debe tomarse en cuenta que hay tweets que hacen referencia a más de una acción, en estos casos se los ha tomado en cuenta en cada dataset de cada acción.

5.2.2. Extracción de atributos con sentiment analysis

Para realizar el sentiment analysis a cada tweet se ha utilizado la librería NLTK de Python, específicamente el módulo nltk.sentiment.vader, que es la implementación del algoritmo VADER [21], que es en sí mismo un vocabulario y una librería que incluye puntuación, emoticones, acrónimos y lenguaje informal que abunda en las redes sociales. Este algoritmo utiliza un modelo basado en reglas que da a cada oración una puntuación numérica que está en el intervalo de -1 a 1, indicando la tendencia del sentimiento a ser negativo o positivo. También devuelve puntuaciones independientes de intensidad del texto respecto a si es negativo, neutral y positivo. En la Tabla 10 se muestran cinco tweets de ejemplo aplicados el sentiment analysis con VADER.

tweet	negativo	neutral	positivo	global
$FB Betting it will simply not collapse has been a huge winner with options	0.000	0.534	0.466	0.8284
Amazon Seeks Fresh Investment in India With New Grocery Service $AMZN	0.000	0.692	0.308	0.4588
$SPY : Has the Correction Started? Get Short, Medium & Long Term Analysis	0.000	1.000	0.000	0.0000
$XOM The New Exxon: Fewer Risks, Lower Returns	0.417	0.583	0.000	-0.5106
$SPY new lows, i'm looking for a morning sell off & afternoon bounce today	0.141	0.859	0.000	-0.2023

Tabla 10: Tweets aplicados sentiment analysis con VADER

Como atributos para el dataset de tweets se ha elegido los cuatro valores que devuelve VADER. Sin embargo, para crear este dataset se necesita agrupar los registros en los intervalos de 5, 15 y 30 minutos. En otras palabras, para cada compañía se ha creado tres datasets de los tweets que las mencionan, uno en cada intervalo, que contiene la suma agregada de los valores que devuelve VADER en cada tweet respectivamente. Además, se ha agregado también como atributo el número total de tweets en ese intervalo. En la Tabla 11 se muestra parte del dataset para SPY en el intervalo de 30 minutos. De forma similar que en el dataset de precios de acciones, se tomará como ejemplo solamente SPY, sin embargo la estructura de los datasets de las demás compañías es semejante.

fecha y hora	negtws	neutws	postws	globtws	numtws
2017-04-19 16:30:00	0.9000	8.9830	2.1170	1.7626	12
2017-04-19 17:00:00	1.2300	12.8930	2.8770	1.6133	17
2017-04-19 17:30:00	1.6150	13.0510	2.3340	1.7615	17
2017-04-19 18:00:00	1.2500	6.7720	1.9780	1.1710	10
2017-04-19 18:30:00	0.2250	10.1330	0.6420	0.1055	11

Tabla 11: Estructura del dataset de tweets para SPY

5.2.3. Descripción de la aplicación para extraer tweets en realtime

Para acceder a las APIs de Twitter se debe solicitar las credenciales respectivas, el único requisito es tener una cuenta en la red social. Para obtener tweets en realtime se dispone de una librería llamada Tweepy [46] que implementa de una forma sencilla la conexión a las APIs de Twitter mediante Python. El script creado establece una conexión permanente a Twitter y recibe los tweets para que sean guardados en un archivo de texto que después será procesado desde Jupyter. Tweepy dispone de métodos para solicitar los tweets con todos los filtros que Twitter ofrece en su API que han sido trasladados al script como parámetros.

Se verá más adelante que con Spark no se necesitará de este módulo debido a que la descarga de tweets se la puede hacer en el mismo software.

6. MODELIZACIÓN CON DATOS EN BATCH

En este capítulo se trabaja sobre la primera parte planteada en la metodología que es la modelización con datos en batch. Los resultados servirán para evaluar el impacto que tiene incluir la información de redes sociales en el modelo de predicción de precios de acciones.

6.1. Descripción de los datasets

De la etapa de preprocesamiento se extrajeron seis datasets por cada acción, uno de precios de acciones y otro de tweets, por cada uno de los tres intervalos de tiempo (5, 15 y 30 minutos). En general se seguirá describiendo solamente los que corresponden a SPY en el intervalo que se indique, los de las demás compañías tienen una estructura similar.

6.1.1. Dataset de acciones

Cada dataset de precios de acciones está ordenado ascendentemente según su fecha y hora, y todos contienen registros desde las 09:30 del 9 de marzo hasta las 16:00 del 14 de junio de 2017. En la Tabla 12 se muestra el número de registros que serán empleados para el entrenamiento y la evaluación del modelo para cada una de las compañías.

	train	**test**	**total**
5 min	4266	1106	5372
15 min	1458	378	1836
30 min	756	196	952

Tabla 12: Número de registros para train y test de cada dataset de acciones

Recordando que ‘close’ es precio de cierre de la acción y está medido en USD, ‘volume’ es el volumen de acciones que se comercializaron en ese intervalo de tiempo y esta medido en unidades, ‘hlp’ es la variación entre el precio más alto y más bajo de la acción en ese mismo intervalo y es un porcentaje, y finalmente ‘cop’ es similar al anterior pero entre el precio de cierre y de apertura; en la Tabla 13 se muestran varios estadísticos del dataset de acciones a 5 minutos para el símbolo SPY. Se

aprecia claramente la diferencia entre los valores numéricos de cada una de las columnas, mientras los de ‘volume’ superan las centenas de miles, los de ‘cop’ está muy por debajo de cero. Más adelante se indicará cómo se normalizan estos valores.

	close	volume	hlp	cop
promedio	238.18	211937.10	0.000588	8.426958e-08
desviación	3.14	397772.80	0.000376	4.368615e-04
mínimo	231.82	10375.00	0.000000	-2.719295e-03
máximo	244.96	9677227.00	0.003364	2.545501e-03

Tabla 13: Estadísticos del dataset de precios de SPY a 5 minutos

6.1.2. Dataset de tweets

De forma similar, cada dataset de tweets está ordenado ascendentemente según su fecha y hora y comprende el mismo intervalo que el de acciones. En la Tabla 14 se muestra el número de registros para entrenamiento y evaluación en cada uno de los intervalos.

	train	test	total
5 min	22176	6048	28224
15 min	7392	2016	9408
30 min	3696	1008	4704

Tabla 14: Número de registros para train y test de cada dataset de tweets

Respecto a las unidades de los campos del dataset de tweets, se recuerda que ‘negtws’, ‘neutws’, ‘postws’, ‘globtws’ corresponden a la suma en cada intervalo de la calificación que entregó VADER a cada tweet respecto a si es negativo, neutral, positivo y su calificación global, todos estos campos son números en coma flotante. En cambio ‘numtws’ es la suma de tweets en ese intervalo y es un número entero. En la Tabla 15 se muestra información de estos campos.

	negtws	neutws	postws	globtws	numtws
promedio	0.108791	1.513530	0.181606	0.124794	1.814059
desviación	0.240929	2.184251	0.333409	0.508157	2.613136
mínimo	0.000000	0.000000	0.000000	-3.905700	0.000000
máximo	3.969000	40.232000	4.500000	5.032100	47.000000

Tabla 15: Estadísticos del dataset de precios de SPY a 5 minutos

Respecto al número de tweets por intervalo de tiempo, en la Tabla 16 se muestra información que corresponde a SPY. Se puede afirmar que no existe tweets en todos los intervalos, sin embargo la media de tweets siempre supera la unidad.

	5 min	15 min	30 min
promedio	1.81	5.44	10.88
desviación	2.61	7.00	13.34
mínimo	0.00	0.00	0.00
máximo	47.00	96.00	189.00

Tabla 16: Estadísticos del número de tweets para SPY

6.2. Métrica de evaluación

Se pretende encontrar un modelo que prediga la variación del precio de determinada acción después de un periodo de tiempo, por lo tanto, el modelo será evaluado en base al porcentaje de aciertos sobre el total de predicciones. La métrica que se usará para evaluar los modelos es la exactitud de la predicción. En este caso queda representada por:

$$Exactitud = \frac{Predicciones\,correctas\,(Up\ y\ Down)}{Total\,de\ predicciones}$$

6.3. Modelización

Para establecer un modelo de predicción de la variación del precio de acciones se ha escogido tres de los métodos de clasificación que soportan las librerías de machine learning de Python: Regresión Logística, Linear Discriminant Analysis (LDA) y Support Vector Machines (SVM). Especialmente el primero, debido a que como se verá en el próximo capítulo, es el único que soporta Spark con datos en stream.

A continuación se describen cuatro escenarios en donde se modela con diferentes atributos de ambos datasets, en cada uno se indica la exactitud del modelo en la etapa de entrenamiento que ha sido calculada mediante un proceso de cross validation de diez particiones (k=10). En cada uno se ha normalizado los datos, restando la media y dividiendo para la desviación estándar. Además, y debido a la volatilidad de la variación del precio de acciones, en todos los casos se ha considerado solamente información del último periodo de tiempo para hacer la predicción.

6.3.1. Primer modelo: Solo acciones

En este primer modelo se utilizará solamente la información del dataset de precio de acciones para intentar predecir la dirección de la variación de su precio. Este resultado servirá como base para evaluar la posible mejora si se agregan los tweets o los precios de las acciones más representativas en en el caso de SPY.

En base a los campos del dataset de precio de acciones, los predictores y variable de respuesta para este primer modelo son los siguientes:

- **Predictores:** $close_{SPY}$, $volume_{SPY}$, hlp_{SPY}, cop_{SPY}
- **Respuesta:** $direction_{SPY}$ (Up o Down)

Los resultados que arroja el modelo se muestran en la Tabla 17.

método	dataset	5 min	15 min	30 min
Log Reg	train	53%	52%	49%
	test	54%	53%	47%
LDA	train	53%	53%	49%
	test	54%	53%	47%
SVM	train	53%	52%	54%
	test	54%	55%	55%

Tabla 17: Exactitud del primer modelo para SPY

A simple vista se puede afirmar que el resultado es demasiado bajo, casi igual que lanzar una moneda o incluso peor. Si embargo y como ya se mencionó, los resultados de este primer modelo servirán de referencia para comparar con los siguientes modelos que incorporan más predictores.

6.3.2. Segundo modelo: Acciones más tweets

Para incorporar la información de los tweets al dataset de acciones se debe considerar que el segundo contiene registros solamente en horario de 09:30 a 16:00, lo que genera información ausente de precios de acciones en esos registros. El enfoque que se ha manejado, y que ha dado el mejor resultado es de llenar con ceros, la razón es que en esos intervalos no existieron transacciones. Los campos involucrados para este segundo modelo son los siguientes:

- **Predictores:** $close_{SPY}$, $volume_{SPY}$, hlp_{SPY}, cop_{SPY}, $negtws_{SPY}$, $neutws_{SPY}$, $postws_{SPY}$, $globtws_{SPY}$, $numtws_{SPY}$
- **Respuesta:** direction (Up o Down)

Se debe tomar en cuenta que el dataset con el que se entrena este modelo tiene más registros que el del modelo anterior, por lo que la exactitud se calcula solamente en los registros que corresponden a los días y horas de apertura de la bolsa de valores (09:30 a 16:00). En la Tabla 18 se muestra la exactitud de este modelo.

método	dataset	5 min	15 min	30 min
Log Reg	train	61%	61%	59%
	test	60%	61%	60%
LDA	train	60%	60%	59%
	test	57%	59%	57%
SVM	train	61%	61%	60%
	test	63%	63%	65%

Tabla 18: Exactitud del segundo modelo para SPY

6.3.3. Tercer modelo: SPY más tweets con otras acciones

En un tercer enfoque se intenta predecir la variación del precio de SPY tomando en cuenta la información de las 10 acciones más importantes que se usan para el cálculo del índice S&P500 que es el que sigue el SPY. En este caso se intenta mejorar el modelo entregando más información al algoritmo. Los predictores y respuesta son los siguientes:

- **Predictores:** $close_{SPY}$, $volume_{SPY}$, hlp_{SPY}, cop_{SPY}, $negtws_{SPY}$, $neutws_{SPY}$, $postws_{SPY}$, $globtws_{SPY}$, $numtws_{SPY}$, $close_{AAPL}$, $volume_{AAPL}$, hlp_{AAPL}, cop_{AAPL}, $close_{MSFT}$, $volume_{MSFT}$, hlp_{MSFT}, cop_{MSFT}, $close_{AMZN}$, $volume_{AMZN}$, hlp_{AMZN}, cop_{AMZN}, $close_{JNJ}$, $volume_{JNJ}$, hlp_{JNJ}, cop_{JNJ}, $close_{FB}$, $volume_{FB}$, hlp_{FB}, cop_{FB}, $close_{XOM}$, $volume_{XOM}$, hlp_{XOM}, cop_{XOM}, $close_{BRK.B}$, $volume_{BRK.B}$, $hlp_{BRK.B}$, $cop_{BRK.B}$, $close_{JPM}$, $volume_{JPM}$, hlp_{JPM}, cop_{JPM}, $close_{GOOGL}$, $volume_{GOOGL}$, hlp_{GOOGL}, cop_{GOOGL}, $close_{GOOG}$, $volume_{GOOG}$, hlp_{GOOG}, cop_{GOOG}
- **Respuesta:** direction (Up o Down)

El total de predictores es igual a cuarenta y nueve, cuatro por cada acción más los nueve de SPY del modelo anterior. La exactitud del modelo se muestra en la Tabla 19.

método	dataset	5 min	15 min	30 min
Log Reg	train	64%	63%	64%
	test	65%	62%	65%
LDA	train	60%	63%	62%
	test	57%	64%	65%
SVM	train	61%	61%	62%
	test	62%	63%	61%

Tabla 19: Exactitud del tercer modelo para SPY

6.3.4. Cuarto modelo: SPY más tweets con otras acciones más tweets

En un último modelo se toma en cuenta la información de acciones y tweets de todas las acciones. Los predictores y respuesta son los siguientes:

- **Predictores:** $close_{SPY}$, $volume_{SPY}$, hlp_{SPY}, cop_{SPY}, $negtws_{SPY}$, $neutws_{SPY}$, $postws_{SPY}$, $globtws_{SPY}$, $numtws_{SPY}$, $close_{AAPL}$, $volume_{AAPL}$, hlp_{AAPL}, cop_{AAPL}, $negtws_{AAPL}$, $neutws_{AAPL}$, $postws_{AAPL}$, $globtws_{AAPL}$, $numtws_{AAPL}$, $close_{MSFT}$, $volume_{MSFT}$, hlp_{MSFT}, cop_{MSFT}, $negtws_{MSFT}$, $neutws_{MSFT}$, $postws_{MSFT}$, $globtws_{MSFT}$, $numtws_{MSFT}$, $close_{AMZN}$, $volume_{AMZN}$, hlp_{AMZN}, cop_{AMZN}, $negtws_{AMZN}$, $neutws_{AMZN}$, $postws_{AMZN}$, $globtws_{AMZN}$, $numtws_{AMZN}$, $close_{JNJ}$, $volume_{JNJ}$, hlp_{JNJ}, cop_{JNJ}, $negtws_{JNJ}$, $neutws_{JNJ}$, $postws_{JNJ}$, $globtws_{JNJ}$, $numtws_{JNJ}$, $close_{FB}$, $volume_{FB}$, hlp_{FB}, cop_{FB}, $negtws_{FB}$, $neutws_{FB}$, $postws_{FB}$, $globtws_{FB}$, $numtws_{FB}$, $close_{XOM}$, $volume_{XOM}$, hlp_{XOM}, cop_{XOM}, $negtws_{XOM}$, $neutws_{XOM}$, $postws_{XOM}$, $globtws_{XOM}$, $numtws_{XOM}$, $close_{BRK.B}$, $volume_{BRK.B}$, $hlp_{BRK.B}$, $cop_{BRK.B}$, $negtws_{BRK.B}$, $neutws_{BRK.B}$, $postws_{BRK.B}$, $globtws_{BRK.B}$, $numtws_{BRK.B}$, $close_{JPM}$, $volume_{JPM}$, hlp_{JPM}, cop_{JPM}, $negtws_{JPM}$, $neutws_{JPM}$, $postws_{JPM}$, $globtws_{JPM}$, $numtws_{JPM}$, $close_{GOOGL}$, $volume_{GOOGL}$, hlp_{GOOGL}, cop_{GOOGL}, $negtws_{GOOGL}$, $neutws_{GOOGL}$, $postws_{GOOGL}$, $globtws_{GOOGL}$, $numtws_{GOOGL}$, $close_{GOOG}$, $volume_{GOOG}$, hlp_{GOOG}, cop_{GOOG}, $negtws_{GOOG}$, $neutws_{GOOG}$, $postws_{GOOG}$, $globtws_{GOOG}$, $numtws_{GOOG}$
- **Respuesta:** direction (Up o Down)

El total de predictores es igual a noventa y nueve, nueve por cada acción incluido los de SPY. La exactitud del modelo se muestra en la Tabla 20.

método	dataset	5 min	15 min	30 min
Log Reg	train	62%	60%	64%
	test	62%	63%	62%
LDA	train	60%	63%	62%
	test	57%	63%	65%
SVM	train	62%	64%	63%
	test	60%	63%	63%

Tabla 20: Exactitud del cuarto modelo para SPY

6.4. Selección del modelo

Para elegir el mejor modelo para SPY se compara la exactitud en cada uno de los intervalos y se selecciona el método que tenga el valor mayor. En la Tabla 21 se muestra de forma resaltada los mejores modelos para cada intervalo. Solamente en el caso de los 30 minutos existe un empate y se seleccionará el modelo más simple, es decir el que involucre menos parámetros, esto corresponde a SVM con acciones más tweets, solamente de SPY. El modelo seleccionado para cada intervalo está señalado con negrita en la Tabla 21.

modelo	método	5 min	15 min	30 min
primero	Log Reg	54%	53%	47%
	LDA	54%	53%	47%
	SVM	54%	55%	55%
segundo	Log Reg	60%	61%	60%
	LDA	57%	59%	57%
	SVM	63%	63%	**65%**
tercero	Log Reg	**65%**	62%	65%
	LDA	57%	**64%**	65%
	SVM	62%	63%	61%
cuarto	Log Reg	62%	63%	62%
	LDA	57%	63%	65%
	SVM	60%	63%	63%

Tabla 21: Comparación de la exactitud de los modelos para SPY

A modo de ejemplo y para visualizar el comportamiento de las predicciones se ha seleccionado el día 9 de junio de 2017. En las Figuras 22 y 23 se muestra la comparación entre el valor real y el predicho.

Dos columnas contiguas indican que la predicción se hizo correctamente. Para el primer caso se calcula en ese día una exactitud de aproximadamente el 57% (16/28) frente al 59% calculado anteriormente. En el segundo caso la exactitud es del 71% (10/14) frente al 65% calculado. Tómese en cuenta que los valores de la Tabla 21 son valores promedio.

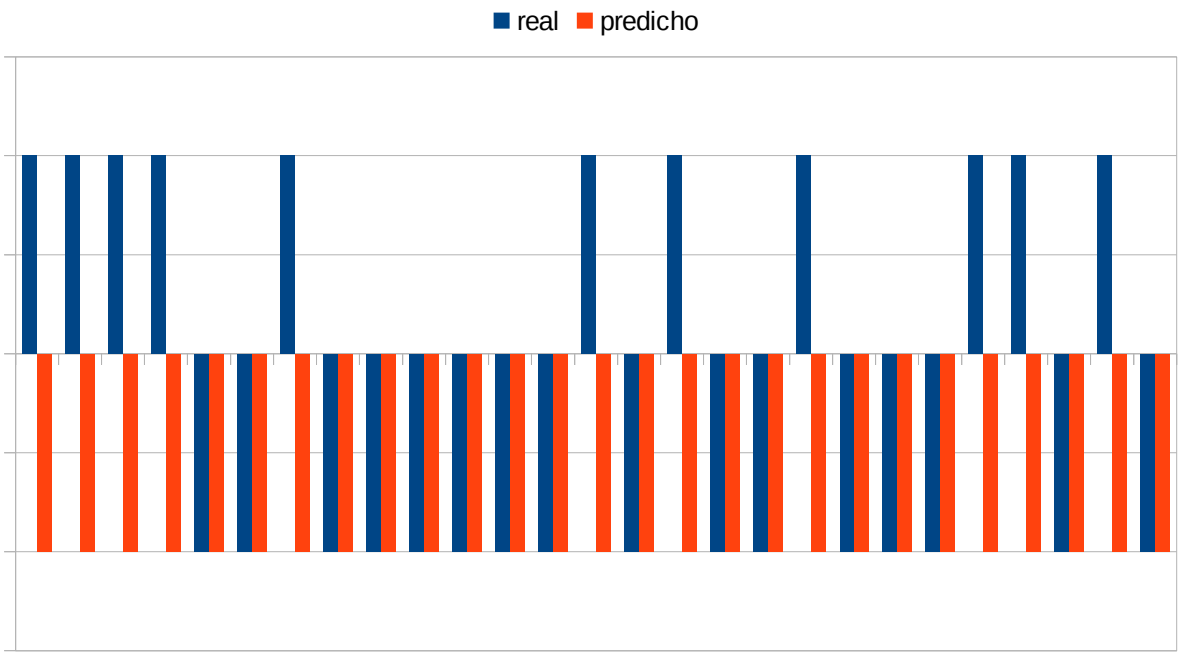

Figura 11: Predicción del segundo modelo para SPY el día 09 de junio (LDA, 15 min)

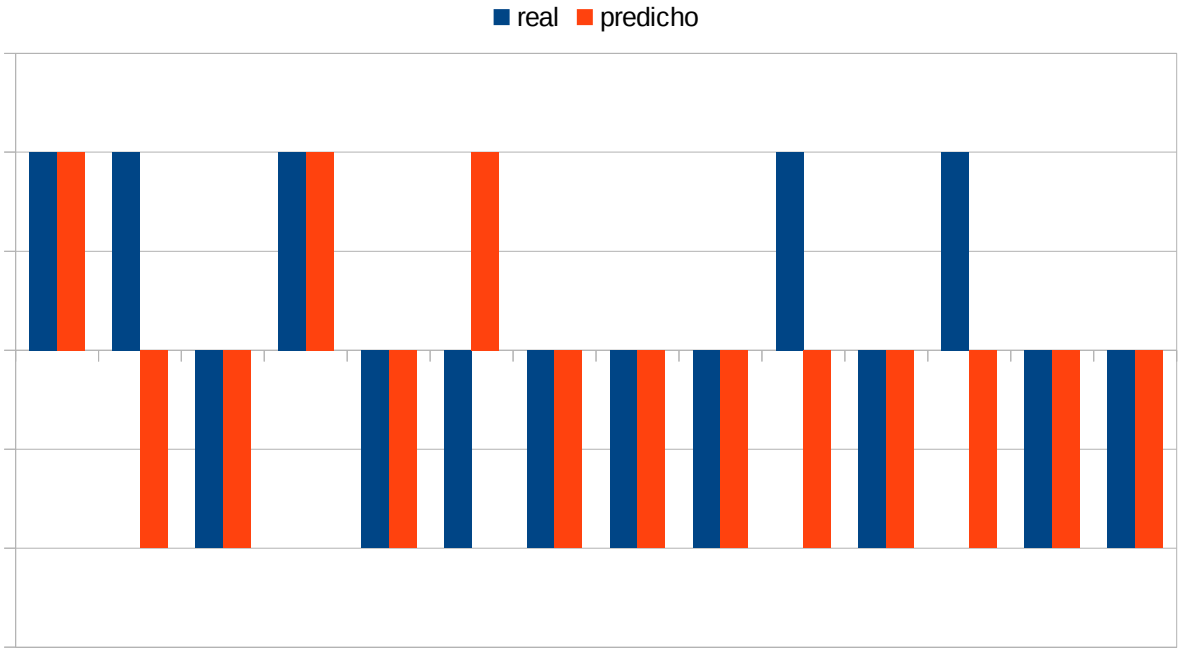

Figura 12: Predicción del segundo modelo para SPY el día 09 de junio (SVM, 30 min)

6.4.1. Modelo para el intervalo de 5 minutos

El método seleccionado es el de Regresión Logística, el cual obedece a la siguiente fórmula:

$$p(direction='Up') = \frac{e^X}{1+e^X}$$

En donde X = 6.552 + 0.112 $close_{SPY}$ + 0.041 $volume_{SPY}$ - 0.147 hlp_{SPY} - 0.143 cop_{SPY} + 0.659 $negtws_{SPY}$ - 0.059 $neutws_{SPY}$ + 0.069 $postws_{SPY}$ + 0.036 $globtws_{SPY}$ + 0.473 $numtws_{SPY}$ + 0.134 $close_{AAPL}$ - 0.018 $volume_{AAPL}$ - 0.015 hlp_{AAPL} - 0.062 cop_{AAPL} - 0.677 $close_{MSFT}$ - 0.004 $volume_{MSFT}$ - 0.046 hlp_{MSFT} - 0.021 cop_{MSFT} - 0.008 $close_{AMZN}$ - 0.128 $volume_{AMZN}$ + 0.043 hlp_{AMZN} + 0.000 cop_{AMZN} - 0.380 $close_{JNJ}$ + 0.004 $volume_{JNJ}$ + 0.052 hlp_{JNJ} + 0.002 cop_{JNJ} - 0.984 $close_{FB}$ - 0.033 $volume_{FB}$ + 0.009 hlp_{FB} + 0.009 cop_{FB} - 0.682 $close_{XOM}$ - 0.040 $volume_{XOM}$ + 0.068 hlp_{XOM} + 0.004 cop_{XOM} + 0.364 $close_{BRK.B}$ + 0.016 $volume_{BRK.B}$ + 0.035 $hlp_{BRK.B}$ + 0.016 $cop_{BRK.B}$ - 1.164 $close_{JPM}$ + 0.031 $volume_{JPM}$ - 0.001 hlp_{JPM} - 0.007 cop_{JPM} - 0.766 $close_{GOOGL}$ - 0.053 $volume_{GOOGL}$ + 0.043 hlp_{GOOGL} + 0.012 cop_{GOOGL} + 0.115 $close_{GOOG}$ - 0.043 $volume_{GOOG}$ - 0.054 hlp_{GOOG} - 0.007 cop_{GOOG}

Si el valor calculado, que es una probabilidad entre cero y uno, es mayor o igual a 0.5 entonces la variable de respuesta del registro X será 'Up', lo que quiere decir que en ese instante la predicción indica que el precio va a subir. Si es menor que 0.5 la respuesta será 'Down'.

6.4.2. Modelo para el intervalo de 15 minutos

El método seleccionado es el de Linear Discriminant Analysis que obedece a la siguiente fórmula:

$$\delta_k(X) = X^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \eta_k$$

En donde:

k = 1 ('Up'), 0 ('Down')

X = vector de atributos para el cual se va a calcular la respuesta.

Σ = matriz de covarianza que es común a las dos clases.

$\boldsymbol{\eta_k}$ = probabilidades a priori de pertenecer a una de las clases.

$\boldsymbol{\mu_k}$ = vector de medias específico por clase.

Los vectores y matrices indicados están en el Anexo 1. Cada registro pertenecerá a la clase cuyo valor calculado en la función δ sea el mayor.

6.4.3. Modelo para el intervalo de 30 minutos

El método para clasificación mediante Support Vector Machines para el intervalo de 30 minutos tiene 352 vectores para 'Up' y 357 para 'Down'. En el Anexo 2 se muestran los índices de estos vectores.

6.5. Resultados y evaluación

En esta sección se analizará uno de los principales objetivos del proyecto, que es verificar el impacto de incluir información de redes sociales en el modelo clásico de relación entre acciones e índice bursátil. Para esto los resultados se dividen en dos casos: según el método de machine learning usado y según el intervalo de tiempo. Todos los gráficos se han elaborado tomando en cuenta la exactitud del modelo cuando se aplica al dataset de testing.

6.5.1. Mejora en la exactitud del modelo según el método utilizado

En la Figura 13 se muestran los resultados obtenidos al incorporar mensajes de Twitter en los cuatro modelos para SPY con SVM. Los resultados con los demás métodos son similares.

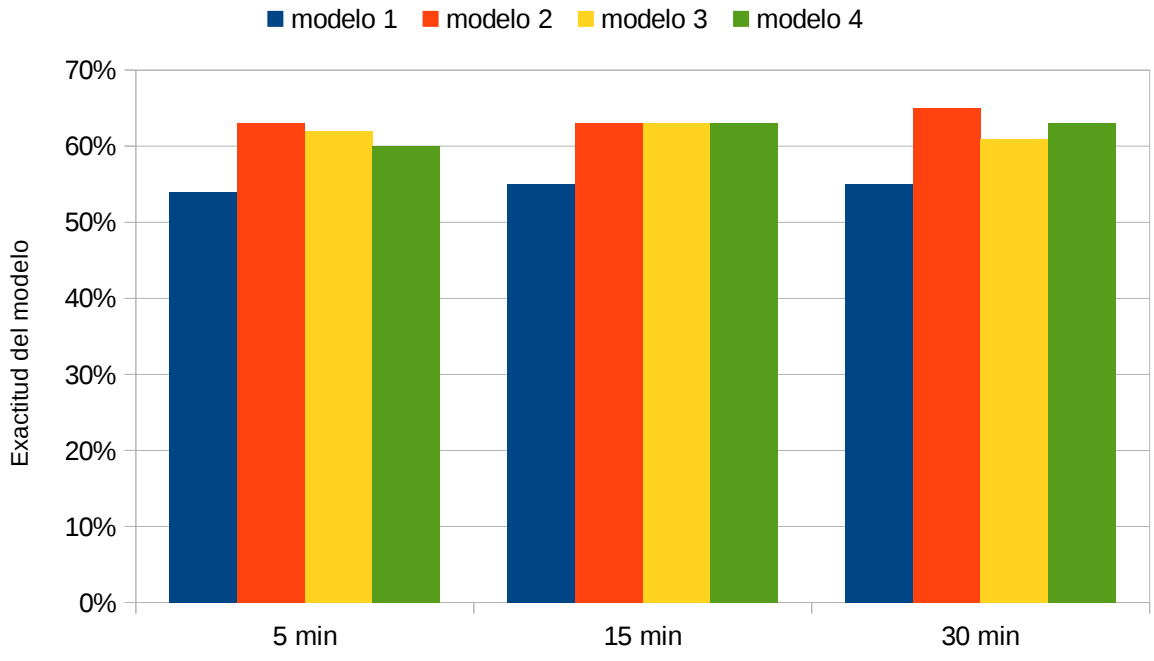

Figura 13: Variación de la exactitud en los cuatro modelos para SPY (SVM)

La exactitud del modelo decrece o se mantiene igual si se incorporan las acciones de las demás compañías, por lo que resulta mejor el modelo con la información solamente de SPY. Se ha obtenido un incremento promedio de 11,67% si se incorporan tweets al modelo.

Respecto a las demás acciones, se analizaron dos casos: solo acciones y acciones más tweets. En las Figuras 14, 15 y 16 se muestran los resultados en el intervalo de 15 minutos. El resultado es que claramente se incrementa la exactitud del modelo en todos los casos a excepción de MSFT con Regresión Logística y LDA.

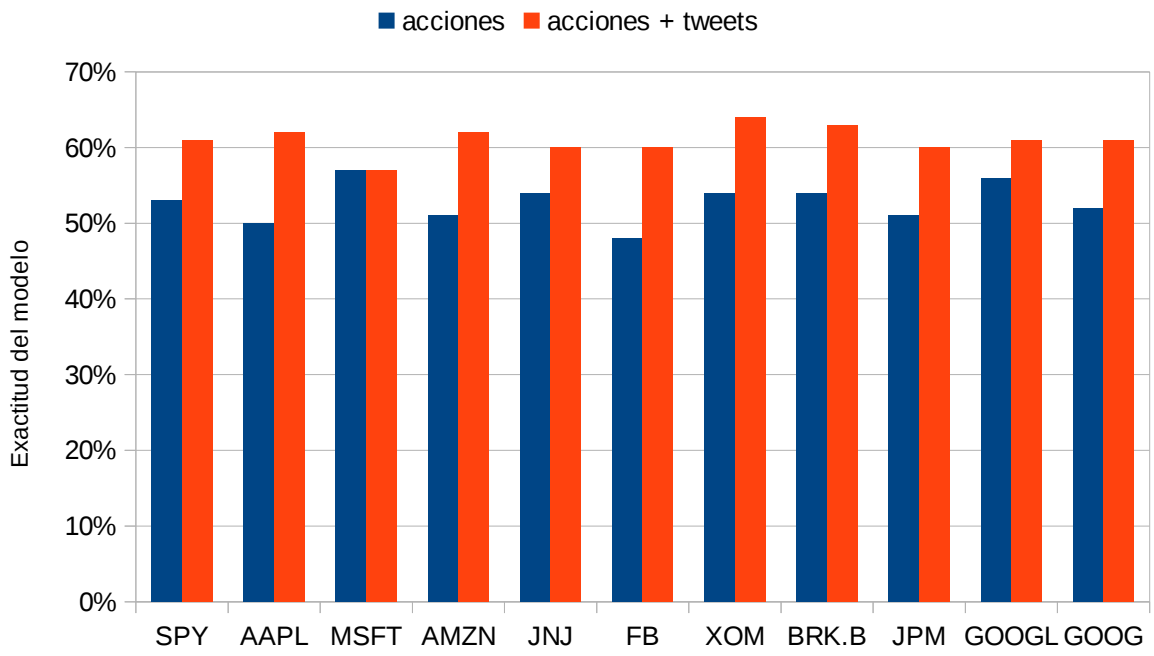

Figura 14: Mejora en la exactitud del modelo con Regresión Logística (15 min)

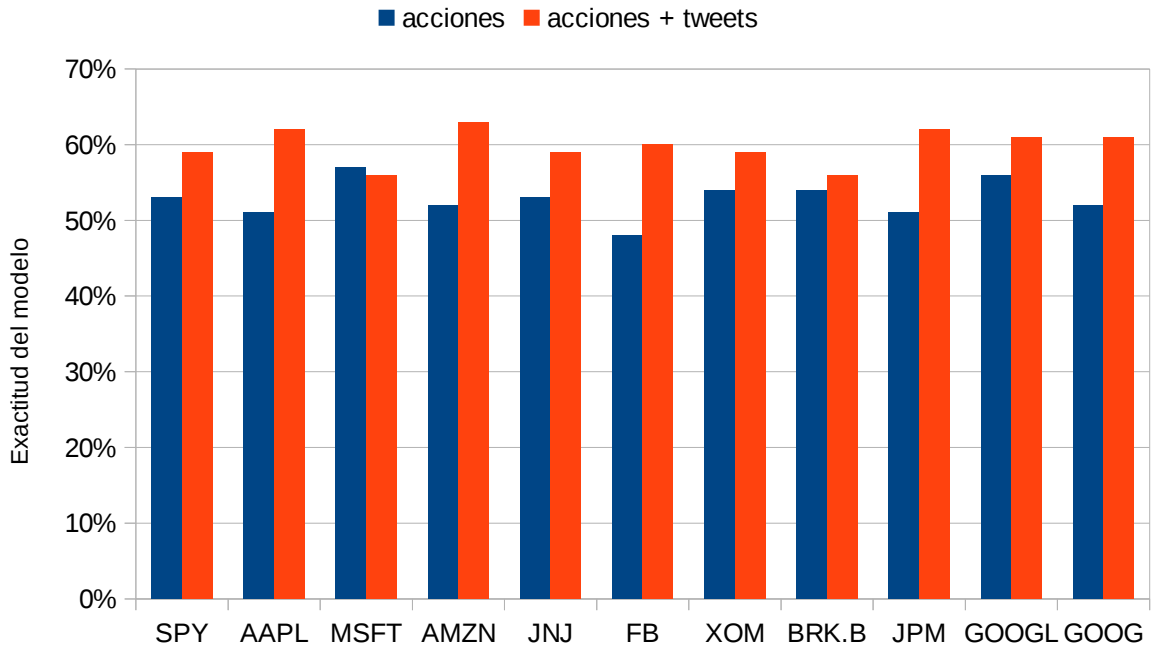

Figura 15: Mejora en la exactitud del modelo con LDA (15 min)

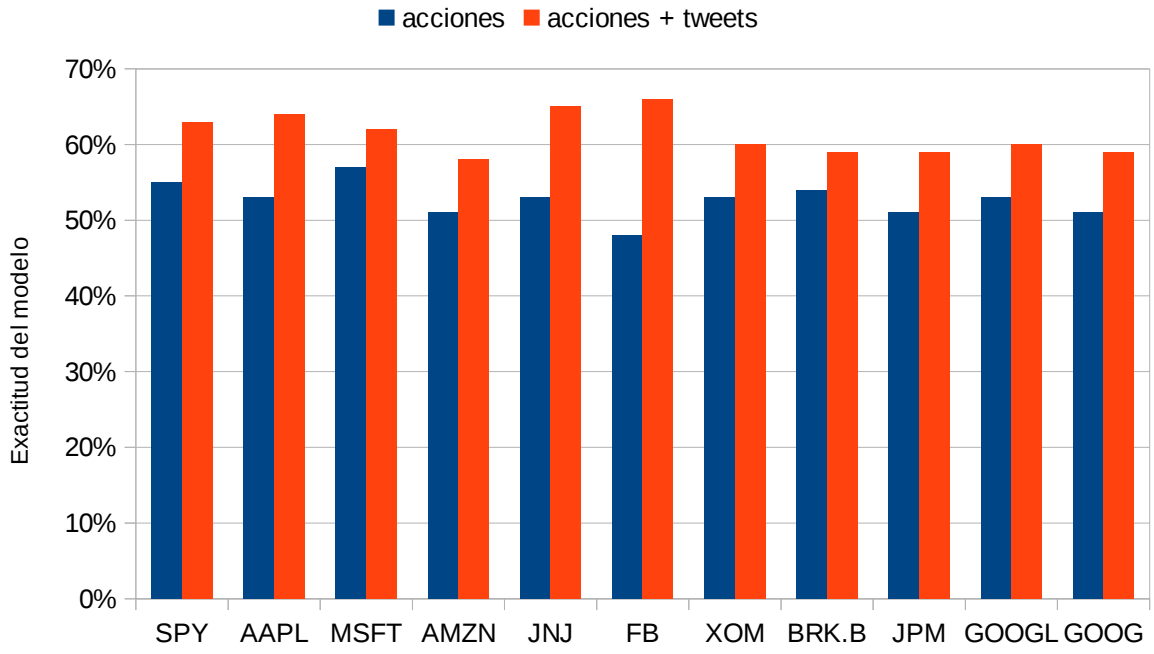

Figura 16: Mejora en la exactitud del modelo con SVM (15 min)

En la Tabla 22 se muestra un resumen del incremento en la exactitud del modelo para cada uno de los intervalos y métodos de machine learning utilizados, en cada intervalo se ha resaltado el mayor incremento de color azul y el mayor incremento promedio de color amarillo.

intervalo	parámetro	Log Reg	LDA	SVM
5 min	máximo	11.00	10.00	13.00
	promedio	7.09	5.82	9.73
	mínimo	1.00	1.00	6.00
15 min	máximo	12.00	12.00	18.00
	promedio	8.27	7.00	8.73
	mínimo	0.00	-1.00	5.00
30 min	máximo	13.00	14.00	12.00
	promedio	10.36	8.09	8.27
	mínimo	6.00	4.00	-1.00

Tabla 22: Resumen de la mejora en la exactitud por modelo

Se puede concluir que SVM se comporta mejor en los intervalos de 5 y 15 minutos, mientras que Regresión Logística en el de 30 minutos, considerando los valores medios.

6.5.2. Mejora en la exactitud del modelo según el intervalo de tiempo

Se consideraron intervalos de 5, 15 y 30 minutos debido a la volatilidad de los precios de las acciones en periodos cortos y también a la variación de la cantidad de tweets en esos intervalos. El resultado de aplicar el modelo con SVM para todas las acciones se muestra en la Figura 17. Con los demás métodos se tiene resultados parecidos en cuanto a su variabilidad.

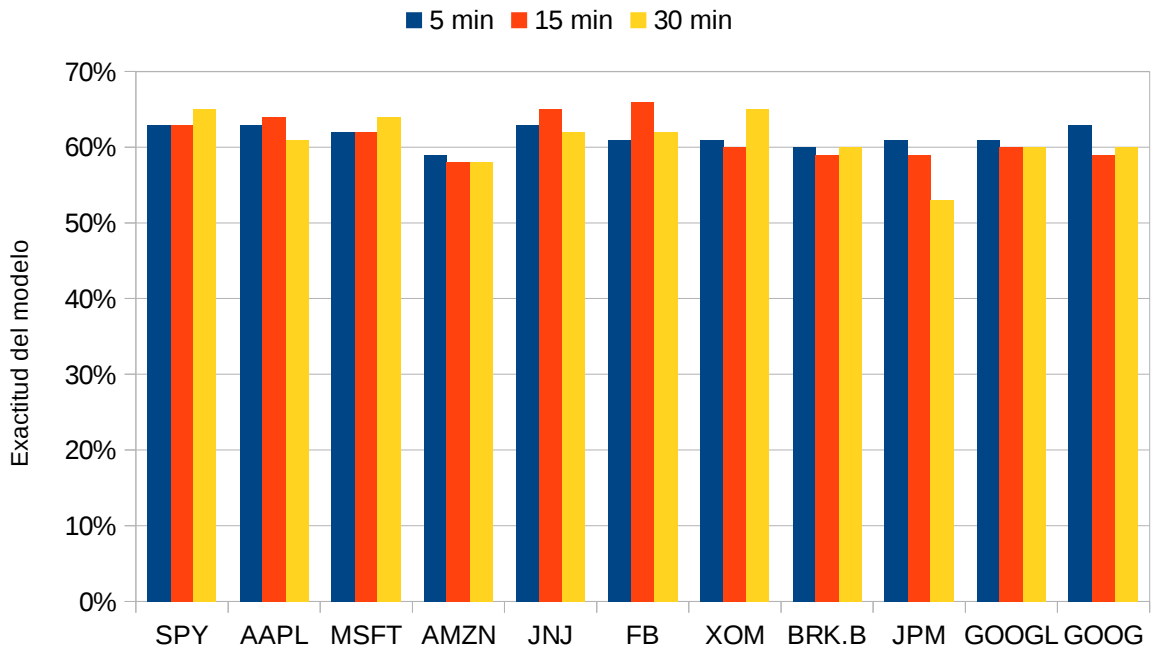

Figura 17: Exactitud del modelo según el intervalo de tiempo (SVM)

A simple vista no se podría afirmar que es mejor trabajar en un intervalo de tiempo específico debido a que el método genera modelos que se comportan de forma diferente para cada acción. En la Tabla 23 se resume el mejor método para cada una de las acciones, se ha incluido su exactitud y en todos los casos corresponde al modelo de acciones más tweets.

	AAPL	**MSFT**	**AMZN**	**JNJ**	**FB**	**XOM**	**BRK.B**	**JPM**	**GOOGL**	**GOOG**
5 min	SVM	SVM	LogReg	SVM	SVM	LogReg	LogReg	LogReg	SVM	SVM
	63%	62%	60%	63%	61%	62%	62%	63%	61%	63%
15 min	SVM	SVM	LDA	SVM	SVM	LogReg	LogReg	LDA	LogReg	LogReg
	64%	62%	63%	65%	66%	64%	63%	62%	61%	61%
30 min	SVM	LogReg	LogReg	SVM	SVM	LogReg	LogReg	LogReg	LogReg	LogReg
	61%	65%	66%	62%	62%	67%	64%	63%	64%	61%

Tabla 23: Mejor modelo para cada acción en cada intervalo

7. IMPLEMENTACIÓN CON DATOS EN STREAM

En esta sección se evaluará comportamiento del modelo con los registros de las tres últimas semanas del dataset y para esto se utilizará Spark, que es capaz de procesar flujos de datos en intervalos de tiempo definidos, lo cual se ajusta perfectamente al trabajo que ha realizado hasta ahora. Se utilizará el modelo de los precios históricos de SPY más los mensajes de Twitter. La limitación existente es que no se podrá aplicar los mejores modelos para cada acción, debido a que las librerías de Spark incorporan solamente el método de Regresión Logística para clasificación con datos en stream.

7.1. Diseño

El diseño se divide en dos partes que muestran la arquitectura de la aplicación en ambiente de producción y en ambiente de pruebas respectivamente. En ambos casos la aplicación funcionará haciendo predicciones solamente en un intervalo de tiempo (5, 15 o 30 minutos) y para una acción a la vez.

7.1.1. Ambiente de producción

En un ambiente de producción se configura Spark para que cree cada Dstream en el intervalo de tiempo que se haya escojido. Mientras transcurre cada intervalo se mantiene abierta una conexión con la Streaming API de Twitter y se recibe todos los mensajes que hagan mención a los cashtags de la compañía que se desea analizar. Esto se lo logra gracias a la librería Apache Bahir [47]. Al final de cada intervalo se realizan los siguientes pasos:

1. Consultar a Alpha Vantage los precios de las acciones mediante una solicitud HTTP, los que se reciben en formato JSON. Se calcula los atributos ‘hlp’ y ‘cop’ y se extrae el valor de ‘close’ para que junto al registro de atributos del intervalo anterior se calcule la dirección que tuvo la variación del precio de la acción. Con este registro completo se actualiza el modelo (entrenamiento).
2. Procesar cada tweet con las librerías NLTK y aplicar el algoritmo VADER para extraer los respectivos atributos. Hacer la suma agregada de los mismos y junto a los datos del paso

anterior se tiene listo el registro para hacer la predicción. Este registro también se almacena hasta el próximo intervalo para efectuar el siguiente entrenamiento.

Cada predicción puede observarse en pantalla o se la almacena en disco para después hacer el análisis. Es importante mencionar que antes de iniciar el proceso es preciso entrenar el modelo con datos en batch, esto se lo hace antes de realizar la primera predicción. En la Figura 18 se puede observar el esquema de esta arquitectura.

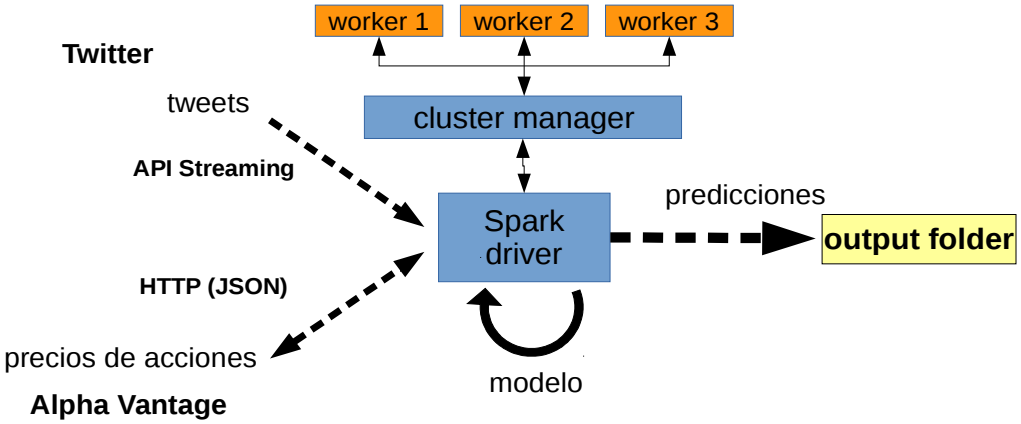

Figura 18: Arquitectura de la aplicación en ambiente de producción

7.1.2. Ambiente de pruebas

Debido a que se planteó evaluar el modelo en tiempo real durante tres semanas, resulta poco eficiente hacerlo con la arquitectura descrita en apartado anterior, por lo que se optó por crear un ambiente de pruebas que optimice el tiempo en verificar la evolución del modelo para cada acción. Además, se dispone del dataset ya preprocesado. La arquitectura utilizada es la que se muestra en la Figura 19 y lo que se hizo es crear un script para que periódicamente entregue un registro a Spark y se pueda hacer el la predicción y el entrenamiento.

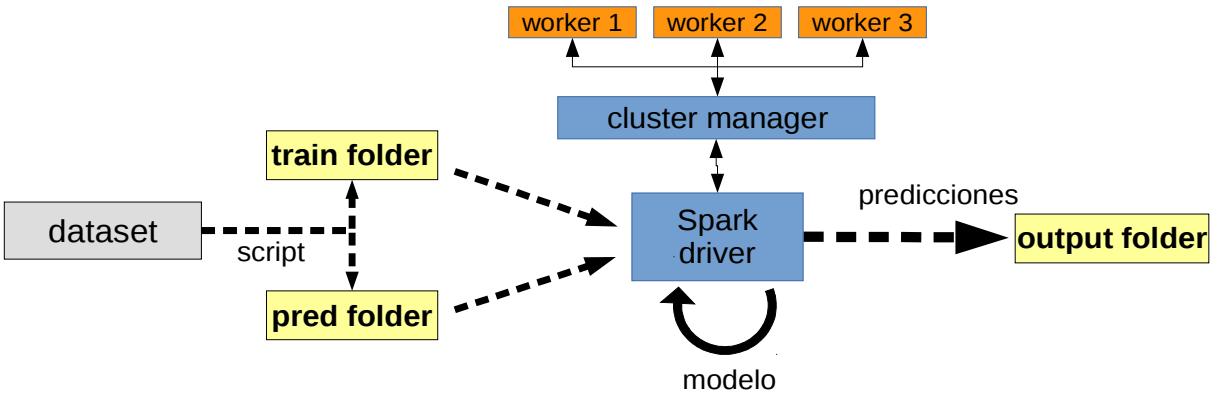

Figura 19: Arquitectura de la aplicación en ambiente de pruebas

Spark tiene la capacidad de crear Dstreams a partir del texto de los archivos que se vayan creando o copiando en algún directorio que se configure. Es así que se han creado los directorios train y predict que se visualizan en la Figura 16, el primero sirve para indicar los datos de entrenamiento y el segundo los datos para hacer la predicción. A diferencia de la arquitectura en ambiente de producción, y debido a que en cada instante se tiene el registro completo sin tener que esperar hasta el próximo periodo para calcular la dirección de la variación del precio, se hace primero la predicción y después el entrenamiento.

Inicialmente se entrena el modelo con la información de los primeros 77 días, posterior a eso se va alimentando registro a registro cada directorio. El objetivo es reducir los 5, 15 o 30 minutos que se necesitaría esperar para hacer cada entrenamiento y predicción, a solamente 3 o 5 segundos, dependiendo del tiempo que a Spark le tome procesar la información.

7.2. Formato de los datos

La salida de Spark se la obtiene mediante un archivo de texto por cada intervalo de tiempo, es decir un archivo con una sola línea en donde se guarda el identificador del registro (que puede ser la fecha y hora) más la predicción. Para clasificaciones binarias Spark soporta como etiquetas solamente los valores 0.0 o 1.0, así que se ha hecho corresponder 0.0 con ‘Down’ y 1.0 con ‘Up’.

La forma de entregar cada registro a Spark es mediante un LabeledPoint, que es una estructura de datos similar a un vector con clave. En la Tabla 24 se muestra la información que se envía en un instante. El primer registro sirve para hacer la predicción, debido a eso se envía como clave un identificador (25), el que es devuelto con la predicción como se muestra en el último registro. El segundo registro, que sirve de entrenamiento, lleva como clave el valor de la predicción, que en segundo registro es el ‘1’ que se muestra al inicio.

directorio	labeled point
pred	(25, [2.08612939858, 0.0745369324881, 0.777677131644, 1.30376588674, -0.404025507998, 2.61544860995, 0.24376588674, 0.34525507779, 3.04544860885])
train	(1, [2.08612939858, 0.0745369324881, 0.777677131644, 1.30376588674, -0.404025507998, 2.61544860995, 0.24376588674, 0.34525507779, 3.04544860885]);
output	(25.0, 1.0)

Tabla 24: Información que se envía y se recibe de Spark

Los dos primeros registros de la Tabla 24 muestran entre corchetes ejemplos de predictores: close, volume, hlp, cop, negtws, neutws, postws, globtws, numtws . Estos se muestran como números en punto flotante debido a que fueron previamente normalizados.

7.3. Funcionamiento

Al dataset ya preprocesado como se hizo en el capítulo cinco, se lo transforma al formato indicado en la sección anterior y se lo guarda en un archivo de texto, para que el script lo lea línea a línea y vaya guardando en archivos en los directorios correspondientes. Se configuró por una parte Spark y por otra el script que fue escrito en Python para que cada tres segundos procesen un registro. Una vez que se haya iniciado Spark se envía primeramente el dataset de entrenamiento, posterior a eso se deja que el script procese cada registro en cada intervalo. Una vez que se procesaron todos los registros, y a través de otro script, se unifica las predicciones que hizo Spark, para luego compararlas con los valores reales. Para SPY se reportó la exactitud en la predicción que se muestra en la Tabla 25.

exactitud	**5 min**	**15 min**	**30 min**
semana 1	60%	56%	54%
semana 2	63%	60%	60%
semana 3	39%	57%	60%

Tabla 25: Exactitud del modelo de SPY en tiempo real

7.4. Resultados y evaluación

Los resultados y la evaluación se efectúa en tres partes, en la primera se evalúa la exactitud en la predicción del modelo con datos en stream, en la segunda parte se compara con los modelos con datos en batch, y en la tercera se evalúa brevemente la aplicación.

7.4.1. Evaluación del modelo con datos en stream

Como se mencionó anteriormente, la limitación que se tiene para entrenar el modelo con datos en stream es que en sus librerías de machine learning solo se dispone del método Regresión Logística. Para verificar la ventaja de actualizar el modelo en tiempo real se entrenó el modelo con los registros de los primeros 77 días y se empezó a predecir y a actualizar el modelo con cada nuevo registro en los siguientes 21 días. En la Figura 20 se muestra el resultado para SPY.

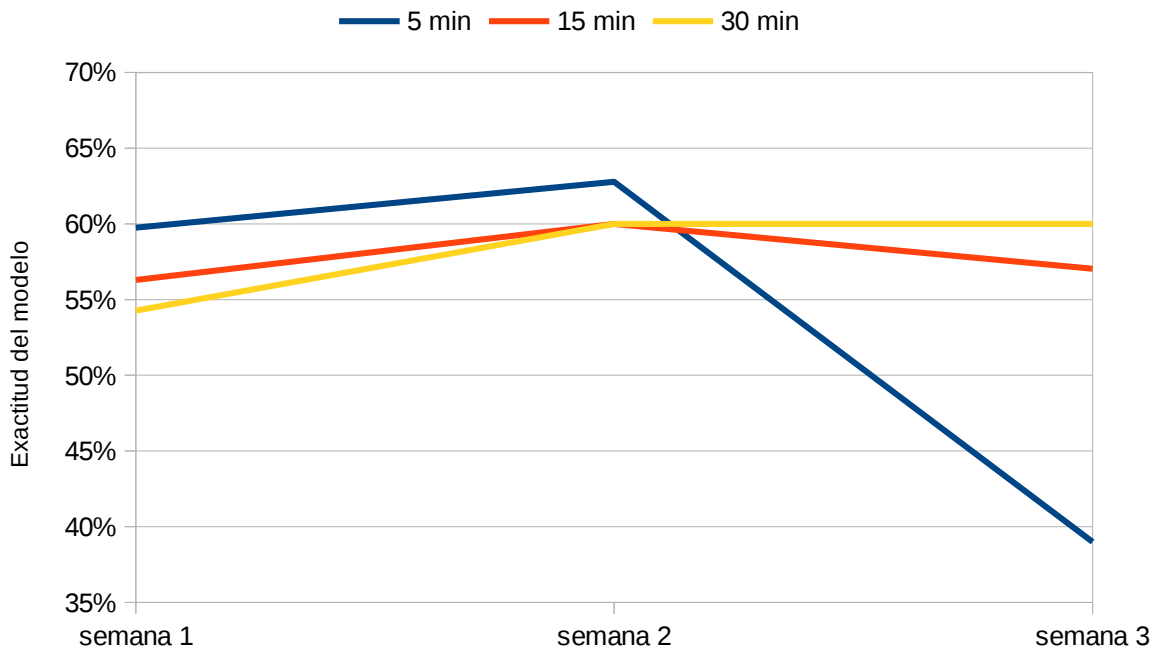

Figura 20: Variación de la exactitud del modelo con datos en stream para SPY

Para SPY se observa un comportamiento no esperado, en el periodo de 5 minutos la exactitud baja considerablemente en la tercera semana. Antes de emitir alguna conclusión se presenta los resultados para las demás acciones en las Figuras 21, 22 y 23, para cada intervalo de tiempo de forma separada.

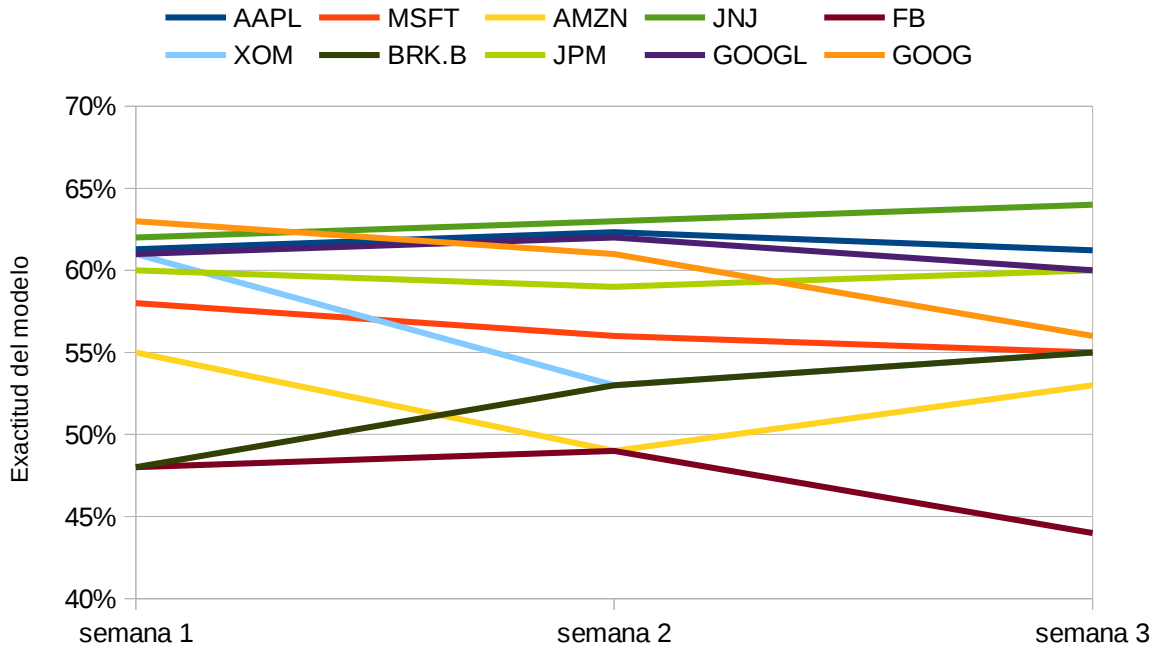

Figura 21: Variación de la exactitud del modelo con datos en stream (5 min)

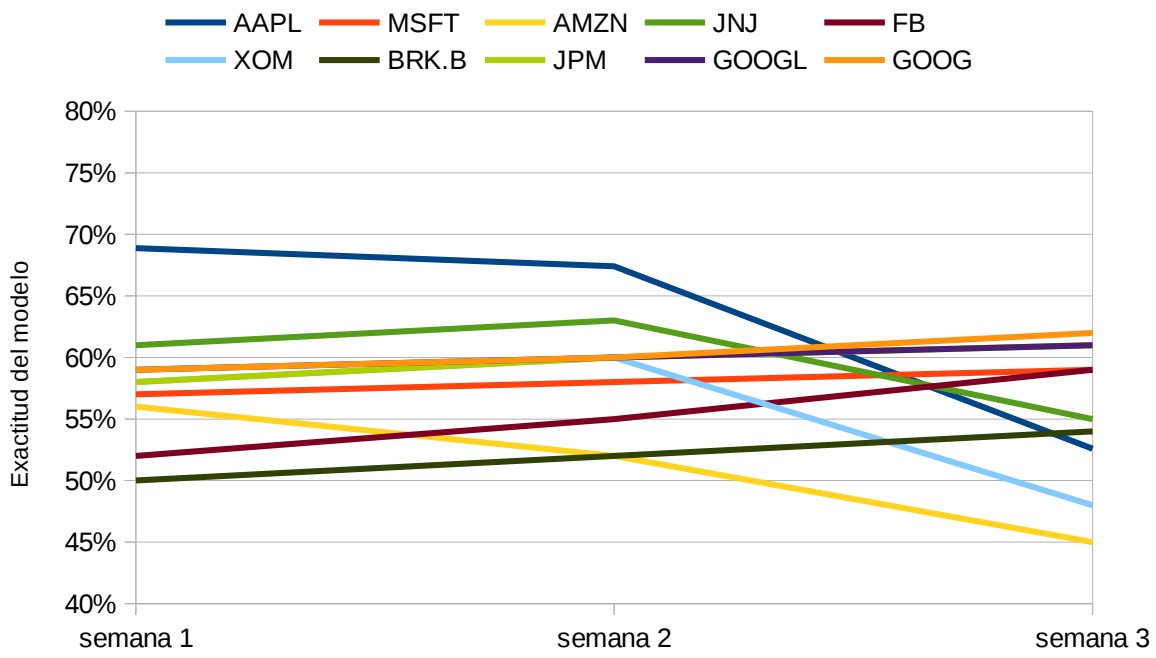

Figura 22: Variación de la exactitud del modelo con datos en stream (15 min)

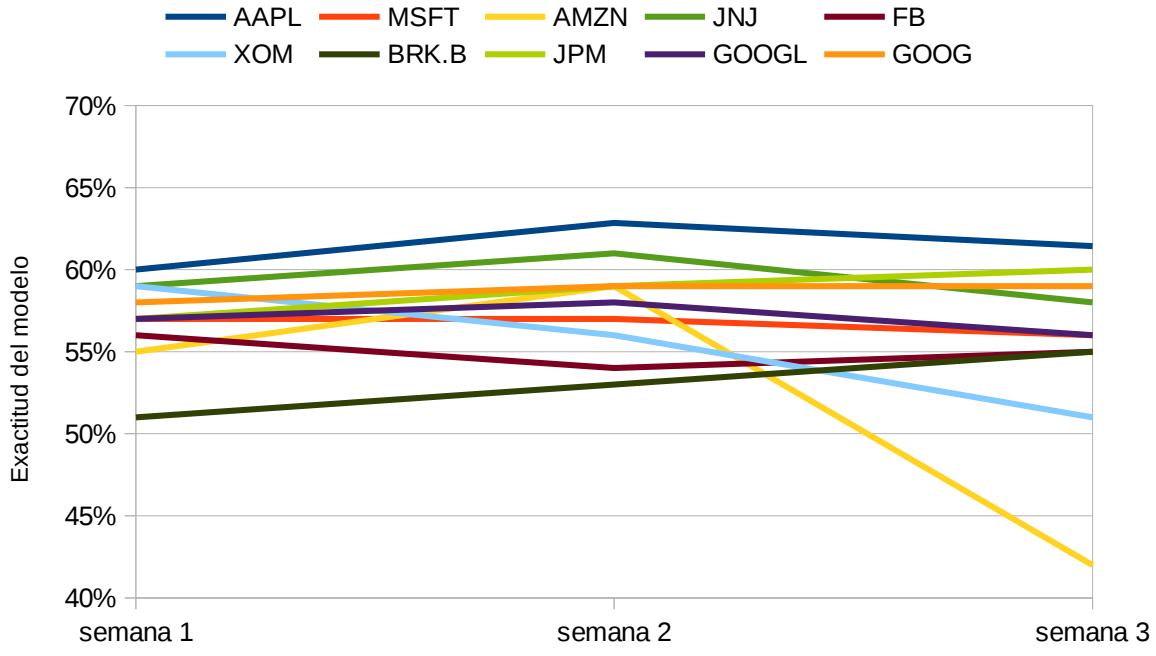

Figura 23: Variación de la exactitud del modelo con datos en stream (30 min)

Son pocos los casos en los que la exactitud aumenta según pasa el tiempo como se esperaba. Se debe tomar en cuenta se tiene pocos datos para entrenamiento y para testing como hacer generalizaciones a partir de estos resultados.

7.4.2. Comparación con el modelo con datos en batch

Para comparar la exactitud del modelo obtenido con datos en stream con la de datos en batch se ha elaborado la Tabla 26. La columna ‘stream’ se ha calculado como el promedio de los valores de las tres semanas que se evaluó. Se puede concluir fácilmente que actualizar el modelo en tiempo real ha provocado resultados más bajos.

acción	intervalo	stream	batch	diferencia
SPY	5 min	54%	63%	-9%
	15 min	58%	63%	-5%
	30 min	58%	65%	-7%
AAPL	5min	62%	63%	-1%
	15min	63%	64%	-1%
	30min	61%	61%	0%
MSFT	5min	56%	62%	-6%
	15min	58%	62%	-4%
	30min	57%	65%	-8%
AMZN	5min	52%	60%	-8%
	15min	51%	63%	-12%
	30min	52%	66%	-14%
JNJ	5min	63%	63%	0%
	15min	60%	65%	-5%
	30min	59%	62%	-3%
FB	5min	47%	61%	-14%
	15min	55%	66%	-11%
	30min	55%	62%	-7%
XOM	5min	56%	62%	-6%
	15min	55%	64%	-9%
	30min	55%	67%	-12%
BRK.B	5min	52%	62%	-10%
	15min	52%	63%	-11%
	30min	53%	64%	-11%
JPM	5min	60%	63%	-3%
	15min	60%	62%	-2%
	30min	59%	63%	-4%
GOOGL	5min	61%	61%	0%
	15min	60%	61%	-1%
	30min	57%	64%	-7%
GOOG	5min	60%	63%	-3%
	15min	60%	61%	-1%
	30min	59%	61%	-2%

Tabla 26: Comparación de la exactitud entre modelización en stream vs en batch

Las diferencias de los resultados en cierta parte pueden atribuirse a que el modelo de Regresión Logística en Spark Streaming no calcula el primero de los coeficientes del modelo, que es un valor constante y que no depende de los predictores. Además, la interfaz de este método en streaming es limitada y no permite hacer una inspección detallada.

7.4.3. Evaluación de la aplicación

La aplicación que se utilizó en este proyecto es la de ambiente de pruebas. Para empezar, la Tabla 27 muestra las características de hardware que se usó.

característica	descripción
tipo	Computador portátil
procesador	Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz
memoria RAM	12 GB SODIMM DDR3 1600 MHz
disco duro	240 GB SSD SATA 6.0 Gb/s

Tabla 27: Características del hardware utilizado

El cluster de Spark se lo implementó con Docker, y aunque no era estrictamente necesario hacerlo debido al poco procesamiento y a que todos los contenedores estaban en el mismo computador, se lo probó de esta manera para verificar el funcionamiento de la aplicación en este ambiente.

El tiempo aproximado que tarda la ejecución de cada uno de los datasets que comprende las tres semanas se detalla en la Tabla 28. Tómese en cuenta que se eligió procesar un registro cada tres segundos debido a que en tiempos más cortos empezaba a haber problemas con Spark y terminaba por abortar la ejecución. Se atribuye este problema a que la ventana de tiempo configurada en Spark era mayor al tiempo que necesitaba para hacer el procesamiento. Además, se necesitaba dejar un retardo entre el entrenamiento y la predicción, que era de dos segundos.

intervalo	tiempo
5 minutos	05:02:24
15 minutos	01:40:48
30 minutos	00:50:24

Tabla 28: Tiempo de procesamiento por el dataset de cada intervalo

Bajo estas condiciones el computador no mostró sobrecarga de procesamiento, la ejecución transcurría sin problema mientras se lo utilizaba para otras tareas.

Respecto a la interfaz de salida, generalmente se guarda los resultados en archivos para después ser procesados, sin embargo se los puede imprimir en pantalla. En la Figura 24 se muestra las predicciones que va haciendo Spark cada tres segundos. De cada par de números de los paréntesis, el primero es el identificador del registro y el segundo la predicción (1.0 o 0.0).

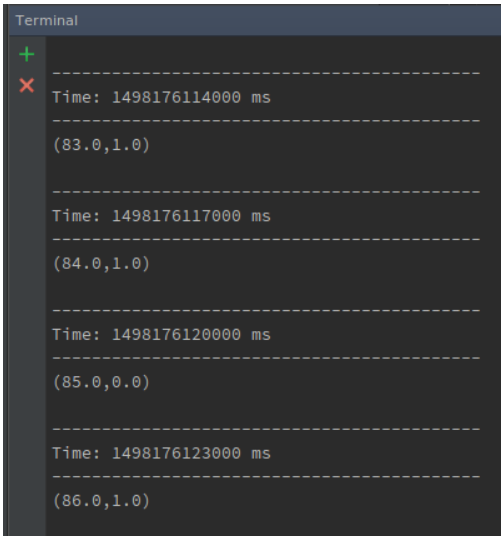

Figura 24: Consola en donde Spark muestra las predicciones

8. CONCLUSIONES Y TRABAJO FUTURO

La culminación de este proyecto ha sido muy satisfactoria por cuanto se han cumplido todos los objetivos marcados al inicio del mismo, además se ha logrado profundizar en los conceptos y poner en práctica las herramientas de la informática actual, sobre todo en el área del data mining. Las siguientes dos secciones comprenden las conclusiones extraídas del proyecto y el trabajo futuro que queda planteado a partir de este.

8.1. Conclusiones

Primeramente, y haciendo referencia a los objetivos principales del proyecto se concluye que definitivamente mejora los modelos predictivos el incluir información de redes sociales como Twitter. A pesar de que los modelos obtenidos no han generado un resultado como para afirmar que pueden ser usados para la toma de decisiones de inversión, se nota un claro incremento en la exactitud de las predicciones, lo que se convierte en un punto de partida para en lo posterior, incluir datos de otras redes sociales, páginas de diarios y demás fuentes de información que estén relacionadas con la bolsa de valores.

Por otra parte, y contrario a lo que se pensaba, actualizar el modelo en tiempo real ha resultado en un incremento en semanas posteriores solamente para ciertas acciones, en general se mantiene constante o decrece. En esta parte se debe considerar que haría falta entrenar el modelo con más datos para observar si existe una tendencia generalizada en los resultados. Respecto a la comparación con los modelos de datos en batch, se debe tomar en cuenta que se está trabajando sobre distintos métodos de machine learning y además cuando estos han coincidido, sobre distintas implementaciones de Regresión Logística.

Los valores obtenidos de exactitud del modelo han reflejado que no existe un solo método de machine learning que produzca los mejores resultados, ni tampoco un intervalo específico de tiempo para todas las acciones. Cada una refleja valores distintos con con los datasets usados, sin embargo, lo que sí se puede concluir es que se prefiere Regresión Logística y SVM a LDA.

En el proyecto se trabajó con datos semi estructurados (acciones) y no estructurados (tweets), a los cuales se les aplicó distintos métodos de extracción de información. El tema del sentiment analysis es

un área que está en continua evolución y es importante trabajarla minuciosamente ya que depende de muchos factores como el ámbito en el que se trabaja, el idioma, los dialectos y más consideraciones, que tomadas en cuenta pueden arrojar mejores resultados.

En el preprocesamiento de datos en stream se debe considerar que en todos los casos no es factible utilizar las técnicas que se usan con datos en batch. Un ejemplo es cuando se llenan los missings con valores calculados a partir de datos posteriores, que en el caso de streaming estos aún no existen. También se debe tomar en cuenta que los tiempos de espera de los datos deben ser menores a los tiempos de la ventana que se configura en Spark.

Respecto a los métodos para extraer datos desde el Internet se observó una clara disponibilidad de librerías y APIs que facilitan esta tarea, tanto para datos en batch como en stream. Esto ayudó a que se pueda implementar correctamente los módulos necesarios de extracción de datos, sin embargo, la implementación con Python para trabajar en batch respecto a la de Spark para trabajar en streaming es diferente y esto obedece a conceptos de computación distribuida. Lo que se hizo en el segundo caso fue simplemente reescribir el código.

Apache Spark ha sido una herramienta de gran ayuda para la consecución de este proyecto, sin embargo, la carencia de diversos métodos para clasificación con datos en stream no ha permitido comparar correctamente con el escenario de datos en batch. Además, presenta ciertas limitaciones que impiden profundizar en la inspección de los resultados que se obtienen. A pesar de ello, si se amplía el proyecto y se trabaja con muchas acciones bursátiles más la información de varias redes sociales entonces Spark puede que sea la mejor alternativa. Un ejemplo sería el trabajar con las 500 acciones que representan el índice S&P 500.

Python y sus librerías de machine learning y procesamiento natural de lenguaje también han sido importantes herramientas para llevar a cabo la extracción, procesamiento y análisis de los datos. Cabe destacar que existe en Internet una enorme cantidad de documentación sobre este lenguaje y sus librerías.

8.2. Trabajo futuro

El algoritmo VADER que se usó para realizar el sentiment analysis de tweets evitó la difícil y ardua tarea de clasificar manualmente cada uno de ellos, sin embargo, si se desea profundizar en el análisis de información de redes sociales es importante trabajar en la creación de un vocabulario específico del ámbito en el que se trabaje, y eso obligaría al uso de herramientas de procesamiento de lenguaje natural como Freeling u OpenNLP. A la vez, esto podría clasificar mejor los tweets y quizá generar mejores predicciones.

En este proyecto se ha utilizado solamente datos de Twitter, pero es un punto de partida para ir agregando datos de otras redes sociales, noticias y otra información que esté relacionada con la bolsa de valores. En este punto hay que tomar en cuenta que respecto al sentiment analysis, quizá se sobrepase el nivel de oración con el que se trabajó en este proyecto, y en ese caso la herramienta VADER ya no sería factible utilizar.

Respecto a Spark, sería conveniente disponer de más métodos para machine learning en tiempo real con los cuales se pueda obtener mejores modelos y hacer comparaciones más certeras. A pesar de que la interfaz para trabajar con datos en stream sigue siendo un tanto limitada se espera que cada vez se vaya flexibilizando como lo ha hecho los últimos años, después de todo se trata de una tecnología nueva de procesamiento de información.

También es importante diseñar e implementar una aplicación que trabaje al mismo tiempo con más de una acción y más de un intervalo. Esto ahorraría mucho tiempo en la búsqueda de un mejor modelo de predicción en tiempo real.

REFERENCIAS

[1] Cómo afecta la reputación de United Airlines su acción en bolsa. Obtenido de http://www.dinero.com/internacional/articulo/united-airlines-afecta-sus-acciones-con-su-reputacion/244144, el 11 de abril de 2017.

[2] Tucker Balch. 2015. Machine Learning for Trading. Curso online en https://www.udacity.com/course/machine-learning-for-trading--ud501. Udacity.

[3] Ko Ichinose, Kazutaka Shimada. 2016. Stock Market Prediction from News on the Web and a New Evaluation Approach in Trading. 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI). IEEE Xplore.

[4] Girija V Attigeri, Manohara Pai M M, Radhika M Pai, Aparna Nayak. 2015. Stock market prediction: A big data approach. TENCON 2015 - 2015 IEEE Region 10 Conference. IEEE Xplore

[5] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. 2014. An Introduction to Statistical Learning. Springer.

[6] Bo Zhao, Yongji He, Chunfeng Yuan, Yihua Huang. 2016. Stock market prediction exploiting microblog sentiment analysis. Neural Networks (IJCNN), 2016 International Joint Conference. IEEE Xplore

[7] Minh Dang, Duc Duong. 2016. Improvement methods for stock market prediction using financial news articles. Information and Computer Science (NICS), 2016 3rd National Foundation for Science and Technology Development Conference. IEEE Xplore.

[8] Ko Ichinose, Kazutaka Shimada. 2016. Stock Market Prediction from News on the Web and a New Evaluation Approach in Trading. Advanced Applied Informatics (IIAI-AAI), 2016 5th IIAI International Congress. IEEE Xplore.

[9] Vivek Rajput, SarikaBobde. 2016. Stock Market Prediction Using Hybrid Approach. International Conference on Computing, Communication and Automation. ACM New York.

[10] Ian H. Witten, Eibe Frank, Mark A. Hall, Christopher J. Pal. Data MiningPractical Machine LearningTools and Techniques. 4E. 2017

[11] Richard J. Roiger. Data Mining, a Tutorial-Based Primer. 2nd Edition. Chapman & Hall/CRC. 2017.

[12] Gibert, Karina, Miquel Sànchez–Marrè, and Joaquín Izquierdo. "A survey onpre-processing techniques: Relevant issues in the context of environmentaldata mining." AI Communications Preprint (2016): 1-37.

[13] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. 2014. An Introduction to Statistical Learning. Springer.

[14] Gohar F. Khan. Seven Layers of Social Media Analytics: Mining Business Insights from Social Media Text, Actions, Networks, Hyperlinks, Apps, Search Engine, and Location Data. Gohar F. Khan. 2015.

[15] Bing Liu. Sentiment Analysis, Mining Opinions, Sentiments, and Emotions. Cambridge University Press. 2015.

[16] A.B. Pawar, M.A. Jawale and D.N. Kyatanavar. Fundamentals of Sentiment Analysis: Concepts and Methodology. W. Pedrycz and S.-M. Chen (eds.), Sentiment Analysis and Ontology Engineering, Studies in Computational Intelligence 639. Springer. 2016

[17] Amazon. Amazon.com Inc. Obtenido de https://www.amazon.com/ el 10 de mayo de 2017.

[18] Filipe N Ribeiro, Matheus Araújo, Pollyanna Gonçalves, Marcos André Gonçalves, Fabrício Benevenuto. 2016. SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods. EPJ Data Science.

[19] Padró, Lluís, and Evgeny Stanilovsky. "Freeling 3.0: Towards wider multilinguality." LREC2012. 2012.

[20] Open NLP. The Apache Software Foundation. Obtenido de https://opennlp.apache.org/ el 01 de junio de 2017.

[21] C.J. Hutto, Eric Gilbert. 2014. VADER: A Parsimonious Rule-based Model forSentiment Analysis of Social Media Text. Association for the Advancement of Artificial Intelli-gence.

[22] Jyoti Ramteke, Samarth Shah, Darshan Godhia, Aadil Shaikh. 2016. Election result prediction using Twitter sentiment analysis. International Conference on Inventive Computation Technologies (ICICT)

[23] Kim YB, Park N, Zhang Q, Kim JG, KangSJ, Kim CH. 2016. Predicting Virtual World UserPopulation Fluctuations with Deep Learning. PLoSONE 11(12): e0167153.

[24] May 2017 Monthly Report. World Federation of Exchanges. Obtenido de https://www.world-exchanges.org/home/index.php/statistics/monthly-reports el 05 de junio de 2017.

[25] Stocks Basics: What Are Stocks?. Obtenido de http://www.investopedia.com/university/stocks/stocks1.asp, el 15 de mayo de 2017.

[26] Yahoo Finance. Yahoo!. Obtnido de https://finance.yahoo.com/ el 01 de junio de 2017

[27] Twitter Company Facts. Twitter. Obtenido de https://about.twitter.com/company el 02 de mayo de 2017.

[28] Twitter Developer Documentation. Twitter Inc. Obtenido de https://dev.twitter.com/docs el 10 de mayo de 2017.

[29] Mohammed Jabreel, Antonio Moreno, Assumpció Huertas. 2016. Semantic comparison of the emotional values communicatedby destinations and tourists on social media. Journal of Destination Marketing & Management.

[30] The Python Tutorial. The Python Software Foundation. Obtenido de https://docs.python.org/3/tutorial/index.html el 15 de mayo de 2017.

[31] Scikit Learn. Scikit Learn Developers. Obtenido de http://scikit-learn.org/stable/ el 10 de mayo de 2017.

[32] The R Project for Statistical Computing. The R Foundation. Obtenido de https://www.r-project.org/ el 15 de mayo de 2017.

[33] Python vs. R: The battle for data scientist mind share. InfoWorld. Obtenido de http://www.infoworld.com/article/3187550/data-science/python-vs-r-the-battle-for-data-scientist-mind-share.html el 22 de mayo de 2017.

[34] Jupyter. Project Jupyter. Obtenido de https://www.r-project.org/ el 17 de mayo de 2017.

[35] Natural Language Toolkit. NLTK Project. Obtenido de http://www.nltk.org/ el 17 de mayo de 2017.

[36] Welcome to Apache™ Hadoop. The Apache Software Foundation. Obtenido de http://hadoop.apache.org/ el 17 de mayo de 2017.

[37] Apache Spark, Lightning-fast cluster computing. Apache Spark. Obtenido de http://spark.apache.org/ el15 de mayo de 2017.

[38] Srinivas Duvvuri, Bikramaditya Singhal. Spark for Data Science. Packt Publishing. 2016.

[39] Spark Streaming Programming Guide. Apache Spark. Obtenido de http://spark.apache.org/docs/latest/streaming-programming-guide.html el 15 de mayo de 2017.

[40] Machine Learning Library (MLlib) Guide. Apache Spark. Obtenido de http://spark.apache.org/docs/latest/ml-guide.html, el 15 de mayo de 2017.

[41] Apache Mesos. The Apache Software Foundation. Obtenido de http://mesos.apache.org/ el 17 de mayo de 2017.

[42] S&P 500 TOP 50. S&P Dow Jones Indices. Obtenido de http://us.spindices.com/indices/equity/sp-500-top-50, el 01 de marzo de 2017

[43] Docker Community Edition. Docker Inc. Obtenido de https://www.docker.com/community-edition el 01 de junio de 2017

[44] Alpha Vantage. Alpha Vantage Inc. Obtenido de http://www.alphavantage.co el 17 de mayo de 2017.

[45] John Kordonis, Symeon Symeonidis, and Avi Arampatzis. 2016. Stock Price Forecasting via Sentiment Analysis on Twitter. In Proceedings of the 20th Pan-Hellenic Conference on Informatics (PCI '16). ACM New York, Article 36 , 6 pages.

[46] Tweepy. An easy-to-use Python library for accessing the Twitter API. Obtenido de http://www.tweepy.org/ el 15 de mayo de 2017.

[47] Apache Bahir. The Apache Software Foundation. Obtenido de http://bahir.apache.org/ el 01 de junio de 2017.

ANEXOS

Anexo 1: Parámetros de LDA para el modelo a 15 minutos

Matriz de covarianza (Σ):

[[1.04974643e+00 5.20482180e-01 3.79772052e-01 ..., 1.69569200e-01
-1.72222203e-02 1.90564553e-01]
[5.20482180e-01 9.00617199e-01 6.72084503e-01 ..., 2.11837286e-01
-1.11902161e-02 2.42964117e-01]
[3.79772052e-01 6.72084503e-01 9.66823925e-01 ..., 1.61359638e-01
1.80276832e-02 1.76584330e-01]
...,
[1.69569200e-01 2.11837286e-01 1.61359638e-01 ..., 5.26563030e-01
9.89045583e-04 3.64772531e-01]
[-1.72222203e-02 -1.11902161e-02 1.80276832e-02 ..., 9.89045583e-04
6.78293242e-01 -1.15681717e-02]
[1.90564553e-01 2.42964117e-01 1.76584330e-01 ..., 3.64772531e-01
-1.15681717e-02 5.34789203e-01]]

Probabilidades a priori (η_k):

[0.39862013 0.60137987]

Vector de medias (μ_k):

[[0.69326942 1.00510216 0.7544 0.41654937 1.00907829 1.51008868
1.75036721 0.11303691 2.01774513 1.43478104 1.38831256 0.19329339
2.0045861 1.36236586 1.48986333 0.11543913 1.9972878 1.41266786

1.22982611 0.13766651 1.97613473 1.53881702 1.63102546 0.09837169
2.01159628 1.5240753 1.34524457 0.1003452 2.00195867 1.46063193
1.68161487 0.03315629 2.03600473 1.71424 1.77919737 -0.02735638
2.03100578 1.71170079 1.63962193 -0.0812484 2.04768952 1.33108201
1.34174352 0.17837289 1.96912069 1.32870967 1.31378274 0.17058622
1.96827513]
[-0.00371188 -0.01898605 -0.01983329 -0.01678474 -0.01912133 -0.1531889
-0.17421754 -0.01526503 -0.21784547 -0.16780785 -0.17342621 -0.00889467
-0.21868909 -0.16499641 -0.18710539 -0.01404971 -0.21973577 -0.16862407
-0.160126 -0.00805533 -0.22139479 -0.16783395 -0.16698169 -0.01786832
-0.2185782 -0.17657071 -0.17733309 -0.00697306 -0.21911958 -0.17253599
-0.18959056 -0.00502015 -0.21589585 -0.18018997 -0.18256813 -0.01049468
-0.21648091 -0.18383522 -0.18414986 0.00323019 -0.21452618 -0.15851895
-0.17716053 -0.01093298 -0.22187149 -0.16447355 -0.17576333 -0.01066772
-0.22199358]]

Anexo 2: Parámetros de SVM para el modelo a 30 minutos

Índices de los vectores de soporte

[21 23 24 26 27 28 32 67 68 69 71 73 74 77 80
211 213 214 216 218 222 259 260 264 265 267 269 271 307 309
314 319 320 355 356 357 358 362 365 403 404 405 406 409 411
413 414 415 416 547 549 550 552 554 555 557 559 595 596 597
598 600 601 602 603 607 608 643 644 647 648 651 655 695 698
700 701 702 704 742 744 745 746 747 749 750 752 884 891 892
936 941 943 944 980 982 983 985 989 991 1030 1031 1033 1037 1039
1076 1079 1081 1083 1084 1085 1088 1219 1220 1221 1222 1223 1226 1227 1231
1268 1272 1273 1277 1278 1280 1317 1318 1324 1325 1326 1327 1328 1365 1367
1368 1369 1371 1372 1373 1376 1411 1413 1419 1421 1422 1423 1557 1558 1559
1560 1564 1566 1567 1603 1604 1605 1611 1614 1616 1651 1652 1653 1655 1657
1660 1661 1662 1663 1664 1700 1703 1704 1705 1706 1707 1709 1710 1711 1712
1895 1897 1900 1940 1943 1945 1951 1988 1989 1992 1994 1995 1996 1997 1998

2035 2044 2045 2046 2047 2083 2085 2088 2090 2093 2094 2095 2227 2229 2230
2231 2235 2237 2239 2240 2277 2280 2281 2285 2287 2327 2330 2331 2335 2336
2371 2372 2373 2375 2378 2379 2380 2382 2383 2384 2419 2420 2422 2423 2425
2426 2427 2428 2429 2431 2432 2563 2564 2566 2569 2572 2573 2574 2575 2576
2611 2613 2618 2619 2621 2660 2661 2665 2666 2668 2670 2707 2709 2710 2711
2712 2714 2717 2755 2758 2760 2768 2899 2900 2901 2902 2904 2908 2910 2949
2950 2952 2953 2954 2959 2960 2999 3003 3004 3006 3043 3044 3047 3049 3053
3054 3091 3092 3096 3097 3098 3100 3102 3239 3240 3244 3283 3284 3285 3289
3290 3291 3292 3293 3331 3332 3333 3336 3338 3339 3340 3342 3343 3344 3381
3383 3386 3389 3390 3391 3434 3438 3440 3572 3573 3577 3579 3581 3584 3619
3624 3625 3627 3628 3630 3667 3669 3671 3673 3674 3677 3680 19 22 29
30 31 66 70 72 75 76 78 79 212 215 217 219 220 221
223 261 262 263 266 268 270 308 310 311 312 313 315 316 317
318 359 360 361 364 366 367 407 408 410 412 548 551 553 556
558 572 599 604 605 606 609 645 646 649 650 652 653 654 690
691 692 693 696 697 699 703 739 741 743 751 753 883 886 887
888 889 890 893 894 895 897 931 932 933 934 935 937 938 939
940 942 979 981 984 986 987 988 990 1027 1028 1029 1032 1035 1036
1038 1075 1077 1078 1080 1082 1086 1224 1225 1228 1229 1230 1267 1269 1270
1274 1275 1276 1279 1316 1319 1320 1321 1322 1323 1363 1364 1366 1370 1374
1375 1410 1412 1414 1415 1416 1417 1418 1420 1555 1556 1561 1562 1563 1606
1607 1608 1609 1610 1612 1613 1615 1654 1656 1658 1691 1699 1701 1892 1893
1894 1896 1898 1901 1902 1903 1939 1941 1942 1944 1946 1947 1948 1950 1987
1993 1999 2036 2037 2039 2040 2041 2042 2043 2082 2084 2086 2087 2091 2092
2226 2228 2232 2233 2236 2238 2273 2275 2276 2278 2279 2282 2283 2284 2286
2323 2324 2325 2326 2329 2333 2334 2374 2376 2377 2381 2421 2424 2430 2565
2567 2568 2570 2571 2612 2614 2615 2616 2617 2620 2622 2623 2659 2663 2664
2667 2669 2671 2708 2713 2716 2718 2719 2756 2757 2759 2761 2762 2763 2764
2765 2766 2767 2769 2903 2905 2906 2907 2909 2911 2946 2947 2948 2951 2957
2958 2995 2997 2998 3000 3001 3002 3005 3007 3045 3046 3048 3050 3051 3052
3055 3093 3094 3095 3099 3101 3103 3235 3236 3237 3241 3242 3243 3246 3247
3286 3287 3288 3294 3295 3334 3335 3337 3341 3345 3379 3380 3382 3384 3387
3388 3397 3427 3428 3429 3430 3431 3432 3433 3435 3436 3437 3439 3571 3574

3575 3576 3578 3580 3582 3583 3593 3620 3621 3622 3623 3626 3629 3631 3668 3670 3675 3678 3679]